Artificial Anthropomorphic Intelligence

An Essay

Gerard 't Hooft

Institute for Theoretical Physics
Utrecht University
and

Spinoza Institute
Postbox 80.195
3508 TD Utrecht, the Netherlands

e-mail: g.thooft@uu.nl
internet: http://www.staff.science.uu.nl/~hooft101/

Abstract

An approach and a short discussion are presented towards the production of human ways of thinking in computers.

November 2014

1. Introduction

Constructing computer programs that can argue intelligently in the way humans do, has turned out to be notoriously difficult. The challenge is to write software that makes a computer do something as complex as to conceive a joke, or to produce an original idea for solving a problem. It is hard to imagine that a computer will ever "comprehend" a simple text. This essay is not written to pretend that the author has all the answers to the gigantic obstacles that seem to be barring us from achieving such a goal. Even if we consider examples of tasks that may seem easy to do for 'ordinary' human beings, it appears to be prohibitively difficult to instruct a computer to perform them. The reason for writing this is that the subject seems to be shrouded by mystery and misconceptions. In my opinion, there are many false approaches and assumptions.

Furthermore, I do have ideas for addressing these problems, and I cherish my own opinion as to how the architecture of an intelligent machine should be designed. Anthropomorphic intelligence is computer intelligence that operates the same way as the human brain. If that sounds scary, indeed it is, but I will return to that.

A reasonable approach may seem to be to construct the software needed for playing chess, for solving mathematical equations, or more challenging perhaps, for driving a car. The problem with those examples is that such programs aren't truly intelligent. The chess program will never be able to play checkers, or a game of go; the mathematical program gives an error message as soon as a bracket has not been inserted at the right place, and the automobile will not ask questions in relation to the trip made. I do think it is conceivable that flexibilities of the sort mentioned will be provided for eventually, but it is clear that this will never be taken for true intelligence. Also, it is clear that the amount of work needed to obtain such programs will be gigantic. It does not compare well with the amount of work needed to raise a child, whereas the physical nature of the effort is very comparable: the child's brain is the hardware, parents and elementary schools fill in the software, and irrespective of errors and shortcomings in all of them, an intelligent human being is the end product.

Imagine that we could pre-program a computer in such a way that an elementary school teacher, or a 'parent', explains to it the facts of life, helps it to learn to walk and talk, teaches it to read and write, after which the computer can go its own way, scanning the internet to find whatever it wants to know. Start by limiting ourselves to the basics only, the facts of life that matter. We would like to equip the computer with a webcam, which would require that it should also learn to recognise and "understand" what it sees.

If this would be possible, we could use such devices in probes exploring outer space, rendering these much more agile than the present ones, for instance in landing safely on a terrain whose physical nature is basically unpredictable. An even moderately intelligent computer would be able to notice a boulder or a hole in the ground, recognise what it is, decide that it is better to avoid the obstacle, and land safely the way it was instructed to do.

The process of teaching the basic facts of life to a computer will have to take place only once, or just a few times, after which one could create millions of clones. The question is why such devices are so difficult to fabricate; what stops us from doing this right now?

There are quite a few different approaches and strategies that are being studied. Naturally, one looks at the basic structure of the human brain itself. Its physical structure is like a network of long, threadlike cells called dendrites, and furthermore, the brain contains various distinct parts that appear to be specialised for different tasks, so this is used as clues. One of the themes that will be stressed here is that, although they may well contain important indications, such clues should not be taken too literally.

2. A comparable case: human flight

Just to make a point, let me briefly compare our problem with a similar challenge of over a century ago. At that time, the challenge was to construct a machine enabling humans to fly in the air the way birds do. The problem was that the basic principles of flight were not yet well understood.

There were enough examples of living things that succeeded to conquer the air with relative ease: birds and insects. They clearly show that, if you want to fly, you need wings. The architecture of bird wings was suspected to exhibit important clues. Therefore, it was concluded that, if you want to fly, you need to flap your wings, and these wings need to be covered with feathers.

Only later, it was realised that feathers were unnecessary, the surface of a wing just needs to be smooth. Furthermore, although flapping your wings to generate propulsion is possible, propellor engines are more practical; later they were replaced by jet engines, which make you go even faster.

There are two lessons to be drawn from this example. One is that we should not copy Nature too literally. When investigators appear to mimic the dendrite structure of our brain, and its chaotic way of processing data, using spin glasses and the like, these remind me of the feathers on the earliest flight machines. The other lesson is that, once we succeed in constructing machines that can do the same thing as biological organisms, we will quickly outperform our biological pioneers by sizeable margins. We now fly higher, faster, further, and with heavier payloads than any bird or insect.

The same can be said in other comparable examples. When we make buildings, they can be made an order of magnitude taller and heavier than trees; when we dig tunnels, our excavators go a lot deeper and farther than moles; when we dive, we can go deeper than most fish, and we can travel in outer space, which no other living creature is able to do. Also, our telescopes and our microscopes can see a lot better than any living creature can.

All this leads to a thought: if ever we succeed in creating machine intelligence, we will outperform our own, biological brains by a big margin. In fact, when it comes to straightforward computing tasks, we have already shown that we outperform living creatures by huge margins.

3. Prospects

As already observed, the title of this essay, in combination with the conclusion of the previous section, may sound frightening. Indeed, that is what it certainly is. Intelligent machines will definitely transform human society tremendously, and where this will lead us is unforeseeable. Only science-fiction authors have been speculating about the consequences of true machine intelligence. If machines take over all our thinking, what will our place on this planet reduce to? I think their conclusions were not very realistic, and in fact, they were still too conservative. Most SF authors expect that 'robots' will never reach a truly human level of intelligent thinking. My own estimate is that they will surpass us greatly. Today, the only thing computers do much better than humans is the processing of large amounts of data, with great processing speed. What I am thinking of now is that they will be able to *argue*, also much faster and better than we can, solving problems and suggesting to us how to make sharp decisions, with tremendous foresight.

There is a consolation: our human brains are not only intelligent, they are also structured to fulfil social tasks and to fight for the survival of the individual to whom the brain belongs. These aspects we do not need to copy. They belong to the feathers that we must omit. Just as we are not fearful for our airplanes, digging machines and telescopes, there is no need to fear the thinking machines themselves. What we might have to fear is their owners, just like fearing our enemies who might have planes, and watch us with their telescopes. We don't have to be afraid that our airplanes will lay eggs like birds, but we might have to fear our enemies for their newly acquired enhanced intelligence. Our defence will be our own super intelligent gadgets. It would be absolutely great if also our governments will use thinking machines that guide them better in making political decisions than when they had to rely on their own deductions. Don't worry that computers might not understand our emotions. They will, and in fact they'll understand the deeper origins of our emotions much better than we do ourselves, and they can be instructed to take these into account.

4. Architecture

Now here comes the core of my ideas. *How* should one construct such software? Which aspects of the human brain *are* important to take into account and copy? When I address basic problems in theoretical physics, my usual approach is first to try to understand their most basic aspects. To identify the most basic features of what might be called intelligence, one does have to analyse what goes on in the human brain, disregarding the biological circuitry and the biological nature of the data being processed (the feathers).

When a human addresses any problem that (s)he finds in his/her way, the brain first tries to understand its nature. When and where did we encounter something similar before? What was the solution in that case? More precisely, we encounter a *pattern*. The human's first mental action is to identify all those various instants where a similar pattern was encountered before, and which response in these other instances had been the most appropriate. Compare the expectations of the consequences of our responses with what is desired from us, and then choose the best one. Continue and face your next problem.

In practice, problem solving will amount to hundreds or thousands of such pattern recognising actions, comparisons, storage of the expected results, and decisions about the action to be taken next. All of this sounds to be not so different from the architecture of existing software programs. The most important difference may be that the whole body of patterns and experiences to be stored will be larger and more general than in the usual software programs, which are much more specific. Our best bet might be to build the architecture of our program the same way right from the very beginning. What follows is a very crude scheme. Many revisions will be needed as we continue to make it more sophisticated, but what I want to construct here is the basic idea.

The patterns to be recognised will first come in the form of texts. In humans, this will be *speech*, but in our times it seems more appropriate to put written texts at the beginning, and worry about speech recognition later. Or, after texts we will have sounds, and then come pictures, or images. If we would succeed to have our software *understand* texts, that already would be quite something, so let us start there.

Texts come in hundreds of languages. Some texts will actually be computer software themselves, or *computer languages*. Eventually, we want to cover all of these, but let us start with one language, plain English. Like a parent, we will have to protect our child from input data that will be harmful (that is, in the wrong language).

It would be tempting to tell our computer what to look for ("texts come in sentences, divided by *periods*, sentences are divided into words, separated by *spaces*", etc.), but we can also use this to test our program, to see if the computer can figure out these features by itself. So, our first step will be to instruct our computer to recognise patterns in these texts. First, recognise the letters, there's a limited number of species. Again, the fact that each letter can come as a capital, lower case, script or boldface, could be something we could put in our instructions right away, but we can also see if our computer can figure this out itself; these would be useful exercises.

Recognising regularities in the correlations of two successive letters would be the next step. Eventually, the computer must come to the point that entire words are recognised. It still does not know what these words *mean*, but that will come, just like in a child when it is raised. As a good parent, we might decide to feed our software with a couple of dictionaries. That will teach it that the words it recognised, all come with their own little dictionary text, relating them to other words. Eventually, these will form more patterns. Our program must be able to *recognise* patterns.

Of course the recognition process includes the fact that nouns can be singular or plural, that verbs are conjugated, and so on, and in sentences, words form relationships.

5. Pattern recognition

The reader might still think that the previous part of this text is very vage. How do we recognise patterns? And saying that texts will be "understood" sounds of course even much more outrageous to a software designer. Imagine a human who is only exposed to texts during his entire life. No matter how intelligent, what will the human ever understand? Eventually, therefore, we may wish to add images and sounds, and perhaps other experiences, but these would be next steps of sophistication. First, a fundamental technical question: how does pattern recognition work, and how can it be implemented in a program?

I think intelligence is all about pattern recognition. Our aim should be to store the maximum amount of patterns in the memory base and allow these to be accessible. Beginning with the texts, it may still sound easy. Our first file will be a list of all letters (or bytes) found. The letters must be *ordered* All our program can then do is make an ordered list of bytes, and add information to them. What information? Well, each byte will be associated to a handful of other bytes that are most often associated with it, and, later, a list of words where this byte occurs. The byte is the first type of pattern, and the list is a list of the associations generated by that pattern. The computer will generate a very primitive notion of "understanding" the letters. The list is generated by reading lots of texts, and checking how letters are correlated: which pairs occur often and which pairs don't. A software programmer will know how to generate this list.

Next, we will need lists of letter combinations. To establish the fact that adjacent letters are correlated will not be too hard. If all is well, the *space* will form different kinds of correlations than the other letters, and the program may need to be refined such that this fact is recognised. Pairs of letters, including the spaces, will show characteristics that will be evident in the lists. Eventually, the lists will be augmented with lists of words. This is where the encyclopedia might be added.

At first sight, we may seem to be wasting memory space, but one should realise that a list of all words, a dictionary, still easily fits on a small disk, so no reason to worry, yet. It is however true that we wish to be economic with memory. Of course our system will require a random access memory that is as large as possible, but with efficient compression we should be able to reduce the sizes of our files. This is called "forgetting", and, certainly at the early stages of our "teaching" period, forgetting large amounts of useless data will be an important ingredient of our program.

Thus, our text 'recognition' program has reached the stage of recognising words, and as such of course can still hardly match the simplest spelling correction programs that are standard in many text editors. It was however not our aim to design spelling correctors, but to gain experience in pattern recognition. If we learned how to reach the stage of spelling corrector using only our fundamental pattern recognition strategy, this would be a great achievement. And now, we can try to proceed along this road.

6. Understanding

What was done with letters, now has to be repeated with words. Our system will start to make lists of pairs of words frequently encountered. A concise ordering system is to be devised (not by the programmer, but by the program, so that, later, all kinds of patterns can be ordered). Whatever pattern is seen by our computer when reading a text, is looked up and registered. Lists of words or parts of sentences correlated with these pairs are registered in an ordered way. The ordering is very important; putting parts of sentences in alphabetical order is not a good idea; the ordering system must be immune to small changes in a pattern. It must recognise that patterns are similar or very different. This is what our brains do. Two trees may have all their branches in different places, yet look very much alike; the pattern ordering system must put these two trees next to each other, or just in one single place to save memory, and yet be able to find them back when a third tree is encountered. Clearly, our software designer must work very hard to devise the optimal pattern ordering code; actually, this might be the most crucial element of our entire intelligence architecture. It is what makes pattern recognition notoriously difficult.

One might fear that our lists of combinations of words might quickly overflow our memory space. Again, compression of data will be important. But one can also argue that our lists of combinations of words and sentences will start out not being much longer than the lengths of all texts encountered combined. Thus, thousands of texts can be processed this way. Then however, our subroutine of *forgetting* may have to be called for. We look for word combinations encountered frequently (all entries in our lists contain hit numbers), and we will erase the combinations that appear to be rare or not constructive for our further understanding.

What does "constructive for understanding" mean? We must have some reward system. The program is rewarded when it successfully predicts some features of the text after parts of it have been read. If successful, it adds reward points to those entries of the list that were used, while, when 'asleep', the program decides to erase entries that did not produce any rewards. How much can be erased depends on the size of the memory used and the amount of memory still available.

Being successful in predicting parts of the text is my interpretation of "understanding". At much later stages, one might think of other ways to reward the system, but prediction will always be the prime ingredient.

7. Memory size

How much memory space will we need? The total memory capacity of the human brain is not known very well. Some say it could be as large as 1000 terabytes, but this I find difficult to believe; it would require memory units, or synapses in the brain, each to take a volume much less than a micron across. Could our brain perhaps contain molecular-sized memory units? I find this extremely unlikely. I take it that the total number of active, independently addressable memory sites in the brain represents a memory of the order of one terabyte.

One could consider the question from the other end. How many distinct patterns must we have stored in our brains at all times? An educated Chinese person is said to know a few thousand distinct Chinese characters. Suppose that this is about 0,1 % of his total brain capacity; then this would mean that an ordinary person would need to remember roughly a million distinct patterns. Each of these patterns is associated to a description of other patterns. If these descriptions take in the order of 10 000 bytes each, we still occupy only 10 gigabytes.

The human brain is only a few million years old, which is nothing in evolutionary terms. It is probably still very wasteful. The earliest attempts to make intelligent software will probably be even much more wasteful, but we can learn from our experiences. Much of our efforts will go in the registration and administration of our patterns. What is a pattern?

A pattern is a partly ordered sequence of other patterns. "Partly ordered" means that the order is sometimes important and sometimes not, and this distinction cannot be omitted, if we want to keep the total number of patterns under control. A word is a pattern of letters, where their order is important; a tree is a pattern of branches whose shapes and order are unimportant. If we would forget that the branches need not be ordered and the number of leaves on a branch does not matter, we would need millions of different "patterns" to characterise a tree, whereas we only want to use a handful of them (there are different types of trees).

Actually, rather than a pattern of letters, a word may be represented as a pattern of patterns of letters. We will just keep the most efficient way to characterise a word.

A very important subroutine will be the demand to *order* the entire list of patterns an individual wants to have at his disposal. Our software will attach addresses to the patterns in this order, so that they can be quickly found back. The input data will be arranged in patterns. If it is a text, the pattern recognition algorithm will recognise the words, after which the system tries to recognise the patterns formed by the words. This goes on until the entire text is seen as a pattern of patterns. Reward points come if certain patterns are correctly predicted, and if the text fits in with previous experiences. The final predictions are delivered as output.

I do not want to underestimate the difficulty of programming such procedures efficiently, but my suspicion is that, once we succeed, the same architecture can be used again and again.

Returning to the question of memory size, we can now ask how many pattern recognition acts are needed for making a decision, and how large is our inventory of patterns for daily use. Perhaps, in the above, I severely underestimated the number of steps needed to proceed from the recognition of Chinese characters to the actions needed to lead a successful life, but I suspect that this is not the case. One could ask how many times per day do we give ourselves a 'reward' for making a correct prediction or assumption, and how many of the patterns that we are keeping in our memory are involved in these processes. If this happens 1000 times per day, this means only a total of not more than ten million new patterns in a lifetime. The real number of patterns needed will be a lot less, but in practice the number could also be a lot more, depending of the efficiency of our wiring, a very difficult problem indeed.

Instead, it is more likely that the complexity of the algorithms involved was highly underestimated, and that our first attempts at constructing artificial intelligence along these lines will be vastly more inefficient than necessary; this we will have to pay for both in memory capacity as in processing speed.

We have not yet considered images and sounds, both as input and output. Sounds may be chopped in units that make them resemble texts, but images are structured differently. Our software must also partition images in terms of patterns. Texture plays a role similar to letters in a word; objects in an image are like words, and relative positions of objects are like sentences. From there on, our architecture will be as before, but we do recognise that images will be difficult to handle; they may well take a bigger amount of memory space than texts, even if we find the optimal procedure to obtain patterns out of them.

There is a curious fact about the human brain: images of *faces*, particularly human faces, are known to be processed in specially designated areas in the brain; the exact relative positions of eyes, nose, ears, and mouth, are extremely important. Ideally, our software would figure out all by itself the need of handling faces with special attention, but of course our 'parent' or 'teacher' might decide to help a bit.

8. Outlook

The author suspects that continuing from here is in principle straightforward. Find patterns, then patterns in patterns, patterns in patterns in patterns and so on. List all patterns in an order that is chosen as smartly as possible, putting similar patterns together in super patterns, and such that these are reproducible. The whole idea of listing the patterns is that it should be possible to find them back. Recognising patterns back when they are similar but not identical is the biggest problem. This is actually why we advocated to start with simple texts where letters form the patterns. At some points, letters must be identified that are 'similar' (such as capitals and lower cases of the same letter), and these must be allowed to be interchangeable in a pattern. The programmer can then learn from the successes at this very simple level, to see how best to proceed in more complicated cases.

This concludes my concise description of what I see as a promising general strategy. I find it hard to imagine that the basic architecture of the human brain differs much from this scheme, except for having been polished by millions of years of evolution. Also, of course, our brain has a whole variety of input modes, from visual to sound and touch.

Also, at later stages we should not forget that our brain generates its own thoughts that can be preserved in our lists of patterns. This is what is sometimes called 'consciousness', the ability to reflect on what was perceived, and to register these reflections, together with the impressions from outside.

The power of the proposed scheme may be that once our program is on track, it can continue by teaching itself, so that the degree of intelligence ultimately attained will depend directly on the sophistication of the pattern registration algorithm employed.

It seems that, to reach the point where our machine can self-educate, an obstacle of kinds will have to be overcome; after that, our system will just grow. After having conquered the English language, all other languages will be easy to learn, and after that the other input modes in the form of images and sounds can be included.

The human brain seems to have the advantage of being able to use gigantic amounts of parallel processing, but this may be compensated by computer speed in the modern processors. Thus, internally, our anthropomorphic machine intelligence works very differently from the human brain, but eventually, these differences should be immaterial.

If what we stated here makes any sense at all, we won't have to wait very long for the arrival of true anthropomorphic machine intelligence. The hardware for it is already available.

9. Intelligence as the use of knowledge **(Added January 2017)**

When discussing intelligence, we find ourselves most often confronted with making knowledge-based decisions. The more knowledge we take into account, the more complex the issues are, the more intelligent our decisions and actions will look. A question often asked is: how dangerous would it be to have systems around that, in this sense, are vastly more intelligent than humans as individuals?

The answer is, that this is not dangerous at all, it is only beneficial, but more to the point, we are already surrounded by systems of this sort; they are called 'libraries'.

We register all our experiences, for later use . First we do this just in our own memory; this is a standard component of our own human intelligence. But already for thousands of years, humanity *as a whole* has vastly improved its combined intelligence by conquering the ability to read and write, so that, now, he have available vastly bigger sources of information than out own memories. What is happening today, is that the transfer of knowledge between individuals is becoming faster and more and more efficient. And now, we have the Internet. With that, we now have vast resources of knowledge within the reach of our fingertips. From my perspective, I regard the Internet as the most intelligent system, or being, that is aiding us in all conceivable ways, by storing and delivering information. As for the pattern recognition process, we still have to do this ourselves, but the patterns, bits and pieces of knowledge, are now delivered to us through the Internet. Aided with that, we are becoming much more intelligent than humanity has ever been in its past.

I reached this point of view by reflecting on one of our basic abilities that are at the source of our intelligence: science. Hundreds of thousands of scientists are doing experiments, registering observations, and drawing their conclusions. All results appear, in some form or other, on the Internet. No single human being can oversee the entire volume of known scientific facts. More importantly, no single human being can check the scientific claims and weed out the mistakes, so this is not happening in an air-tight way, but we do have peer review. And now, someting very interesting is happening.

No single human being understands it all. In general, we all have distorted views on those branches of science that we are not directly involved with much. Many people are not involved with any branch of scientific knowledge at all. Therefore, at the personal basis, we all make mistakes and misjudgements. This means that, also in the fields of sciences, the information on the Internet is not perfect. However, it seems that there are cleansing mechanisms. Erroneous, ill-formulated, dubious scence results abound on the Internet, and yet, they seem to do little harm. There is what we call a 'main stream': the filtered basic science knowledge, practically free of errors, but not yet including the latest, speculative but potentially very important ideas. My point is that the Internet seems to be sufficiently intelligent that it autocorrects. Again, this autocorrection mechanism is not perfect. Main stream science may also contain mistakes and shortcomings. Every scientist knows of the examples in his/her field. Yet the Internet as a whole seems to be much more intelligent than *any single human being.* Even scientists like myself sometimes discover that there is already much more known about their own field than they realised earlier. And most often, that knowledge turns out to be correct as well. Thus, yes, we are surrounded, and indeed to some extent controlled, by a superintelligence. It is not dangerous. Dangerous are the users of this source, at some occasions.

Now, particularly the science and technology component of the Internet, is still in its infancy. It is like a newborn child, enjoying the intense variety of impressions, bits of information, that it receives from the world it lives in, having itself be heard whenever something is asked . The child however, is not yet well educated, and can still be tought many new things in order to function even better in it future life. I think that, in the not so distant future, the Internet can be used as an extension of everyboby's intelligent mind, without sitting in the way, let alone forming a danger.

Often, when we are searching for the answer to a question, we punch in some well-chosen words in Google, and collect hundreds of search hits. Weeding these out sooner or later helps us to find what we were looking for. In practice, however, I think this procedure is still very clumsy; this way of answering questions is a great success of 20^{th} century science and technology, but the 21^{st} century should be able to educate the Internet much better. The following is an account of what I think should be done now. I know that, actually, computer programmers are investigating such lines of thought (S. Wolfram, personal communication).

A large number of searches I perform on the intenet are about something I do not understand. Often it is something connected with a new gadget I bought. Say I bought a printer, but the software coming with it is complex, and the manual is incomprehensible. I have a problem: I switched on the printer, I manage to connect it to my computer, but it refuses to print, or it prints in the wrong size, on the wrong pieces of paper, it refuses to do this two-sided, or any such problem.

Today, we search using Google. But what doo you type in? "Help, my printer does not work ...", or something as smart as you can imagine. What does it reply? It gives me hundreds of hits, to chat pages of *other people*, who got themselves stuck with *other problems*, somewhat similar to mine, but rarely the same. Regardless whether I find the same question on a chat page, what I usually do *not* find, is the answer. I get depressed, by seeing all those other problems people have, and seeing only very few answers, mostly wrong.

So, improvement # 1: do not return questions by other people, find the question that looks most like the one asked, And *find the answer.*

This would be a tremendous improvement, but it is not good enough, for several reasons. The answers given are usually something like the following:

- Did you switch on the printer? Switch it on.
- Maybe the paper tray is empty. Then put new paper in.
- Is it connected to your computer? Check your wifi connections.
- Maybe your driver XGPr100.30a needs to be updated. Install the newest update from ....
- Are you working with a Mac? You need to follow the following steps ....
- Is your ink cartridge out of ink? Put a new cartridge in.

Etc.

What is wrong with these answers? They are either irrelevant or they force me to figure out the answers to a lot of other questions. What is the matter with my computer connection? Where do I find that driver, and how do I know it has to be updated? My ink cartridge is full – that wasn't the problem. And so on.

In short, this is still a 20$^{\text{th}}$ century response. Answers with question marks in them are outlawed. My new program, call it "Googleplex", does the following. Googleplex does not ask *me* whether my printer is on, it asks the printer. It checks whether there is paper in the tray, whether that driver has to be updated and reinstalled, it downloads and updates itself, when necessary; it checks the level of the ink cartridges, and so on. *Only if something has to be done that Googleplex cannot do itself, it asks me, as the last resource, to do something*, like putting new paper of the desired size in the tray it just opened for me, like filling in the password that only I know, and for safety reasons is not known to Googlplex, and so on. As for the password, if I lost it, it advises where I may have stored it, and does everything to minimise the trouble this costs me. Maybe, for safety, I did not allow Googleplex to manipulate drivers in my computer. In that case, it might ask my permission, go to the page where I can download that driver update, tell be which button to press, and why it is that I have to do it myself.

This is what should happen in the 21$^{\text{st}}$ century. Imagine that much more products will be on-line. Robots will be able to put coffee in my coffee machine and pour me a cup. Of course there will be bugs in such machines. All I have to do is yell at Googleplex and it will not stop before the cup of coffee is in my hands.

Imagine such a service being available in handsets, watches or whatever. One such service is already available: Global positioning systems combined with route planners and the like, may be regarded as essentially 21$^{\text{st}}$ century technology. Clearly, there is a lot more to come. I suspect that Googleplex can be personalised. We will all have our own Googleplex assistent, who knows his owner better then (s)he knows him/herself. This means that the entire Internet is almost at our fingertips at all times; this way, we can, effectively, enhance our intelligence in daily life.