AI and its Limitations in 2026

James Trevelyan, The University of Western Australia

10th August 2026 (updated October 2026)

Background

I am compiling sections of chapters for a new edition for my 2014 book, The Making of an Expert Engineer" for mid-career engineers to build their knowledge and skills.

This is an introduction to contemporary artificial intelligence for mid-career engineers. At the same time, it does not rely on any specialized engineering knowledge so the article could be helpful for a wider audience as well.

Many readers likely graduated before the November 2022 release of ChatGPT set off the current surge of interest in AI. This article should help readers understand this so-called revolution better. It's important to appreciate the true potential and also fundamental limitations of AI technologies to evaluate the mostly unfounded claims that we see in popular media.

Expert readers: I invite you to send me corrections and improved descriptions, or aspects of the technology that I have missed.

Non-experts: Please feel free to tell me what's unclear or needs more careful explanation.

Large Language Models

The most recent surge of public interest in AI, rightly with considerable apprehension, started with the release of ChatGPT version 3.5 at the end of 2022. The core technology, an artificial neural network (ANN), first became more than a curiosity in the late 1980s.

An ANN consists of a number of nodes arranged in layers.[1] In any layer, each output node is connected to all the input nodes by a weight value such that the output is a function of the input data and the weight values. In the training phase, a statistical calculation adjusts the weight values until a given input data set will generate an approximation of the desired output value for each node.

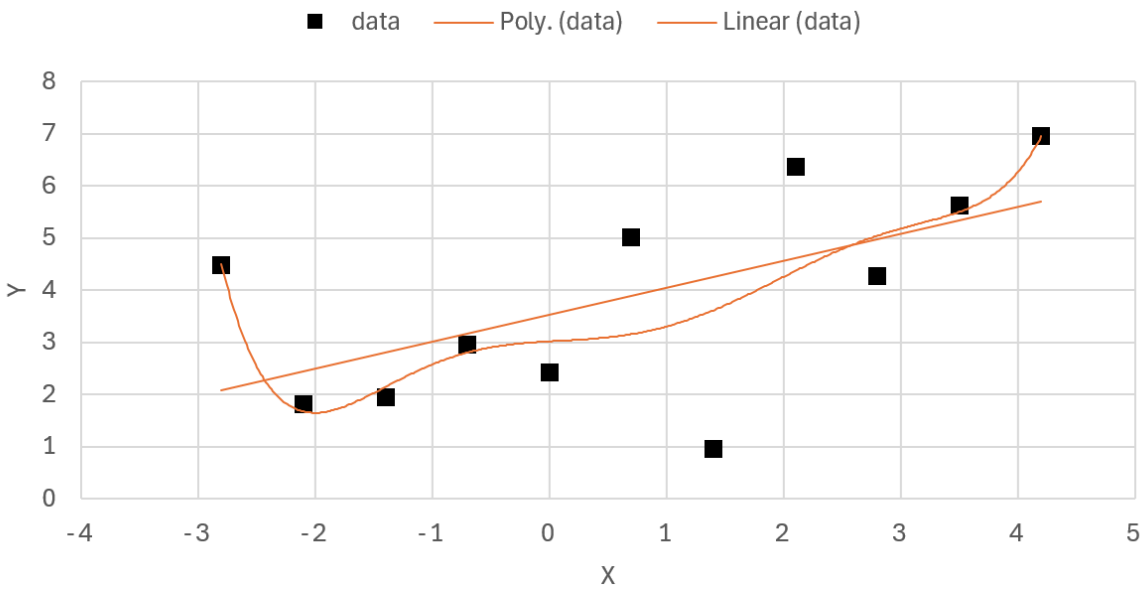
Until the 1980s, attempts to mimic human intelligence with computers had relied entirely on encoding rules used by human experts. While there were some modest successes, it was tediously difficult. An AI specialist had to sit with a human expert in a particular knowledge domain for a couple of years or more, and then the result was a

[1] https://en.wikipedia.org/wiki/History_of_artificial_neural_networks

computer program that had very narrow capabilities. One outstanding success was IBM's Deep Blue super-computer, the first to beat a world grand master in 1996. Actually, Deep Blue did not have a complicated set of rules for winning at chess. Instead, it relied on brute force, the ability to search so many hundreds of millions of possible moves, far more than a human could possible evaluate.

An ANN, on the other hand, does not need input from a human expert. Instead, the statistical process enables the network to "learn" from the training data – inputs and desired outputs. The only pre-determined attributes are the number of interconnected layers and the number nodes in each layer.

In principle, an ANN is similar to fitting a curve to a set of data points with random errors. We can choose to fit a straight line to the data, or a polynomial, as shown in the diagram here. The higher the order of the polynomial, the closer the line will fit the data points. In the same way, an ANN with more nodes may provide a better fit with the input data, depending on what is known about the sources of variability in the input data. Like any statistical calculation, ANNs reflect a choice of assumptions inherent in their design.



While ANNs had some initial application successes with nonlinear control systems, more complex challenges like recognising patterns in images required many nodes and layers. Training calculations were too demanding for computers available at the time for all but the smallest networks.

Modest research efforts through the 1990s and early 2000s gradually yielded improvements that reduced the computing power needed for training. From 2004 onwards, researchers turned to graphics processing unit chips (GPUs) developed for rendering high quality graphics for popular computer games. These chips could handle many computations in parallel, rather than one at a time.

The term “deep learning” appeared among research teams as the number of nodes and layers increased.

The ancient Chinese game Go has become popular across north-east Asia and today internationally. A team of Google researchers managed to train the ANN Alpha-Go to play the game at the highest levels of difficulty. In March 2016, Alpha-Go beat grand master Lee Sedol in a public display of the power of deep learning technology.[2]

A new computational architecture emerged from efforts to apply artificial neural networks to language translation: the transformer. OpenAI popularised the term Generative Pre-trained Transformer (GPT) to describe its own language model. The transformer architecture enabled larger and larger ANNs to be trained enabling the development of statistical models of human languages trained on billions of web pages accessible on the internet. These models became known as large language models (LLMs) and can generate text that can be indistinguishable from human-written text.

Generative AI

Given a text string, an LLM can predict the most likely word to come next. For example, take the following input text string:



The string is divided into tokens shown above with different colours and vertical bars. In this example, the tokens correspond to the words.[3] Longer words may be broken into two or three tokens. The LLM calculates the most likely tokens that follow. For example, Claude.ai gave me a list of the most likely words and phrases that it could expect to follow the string above, even though it cannot actually “see” the probabilities that gave rise to its list.

Next word or phrase	Rough probability estimate
. (period)	0.350
And	0.110
Before	0.080
As	0.060
Then	0.040
While	0.030
before saying	0.020
before adding	0.020
before turning	0.020
That	0.015
When	0.015

[2] https://en.wikipedia.org/wiki/AlphaGo
[3] https://platform.openai.com/tokenizer

and says	0.014
and winks	0.014
and nods	0.013
and turns	0.013
and leans	0.012
and whispers	0.012

Here the most likely token that follows is a period, marking the end of a sentence.

Next, the computer adds this token, a period, to the end of the input text string. Then the LLM repeats the calculation to find the next token, in this case starting a new sentence. Again, the computer adds that token to the end of the input text string, and keeps on repeating these steps until the next most likely token is a "pause", like a break in a human conversation.

A chatbot is a program that enables anyone to access an LLM simply by providing an input text string known as a prompt. After presenting the output text generated by the LLM, the chatbot will pause and invite the user to respond with another prompt. This can lead to an extended dialogue or conversation between the LLM and the user which might last for hours or even several days. For most people, the responses from an LLM will seem uncannily similar to a human response.

OpenAI, founded as a public interest company, released its first GPT on a trial basis in June 2018, but the output text was not coherent. The following February saw the staged release of GPT-2 that could generate coherent text that made sense, albeit with a limited range of responses. Larger and more capable models followed and other companies started to develop their own models.

After the first stage of training using text from the internet, engineers refine the model with a huge collection of training prompts and desired responses. Further statistical calculations adjust the network weights until the LLM responses are judged to be satisfactory. Human testing will follow to make sure, for example, that the LLM will decline to help someone construct a bomb or an infectious pathogen. Extensive testing will precede a public release of any new LLM, and engineers will have to closely monitor real conversations with users to make sure the LLM behaves appropriately.

Today's chatbots have additional features to make them more useful. Without these additions, a text-based GPT might provide complete nonsense.

For example, early LLMs were usually unable to perform simple counting. "How many r's are there in strawberry" was a popular question used to test LLMs which could not count r's to get the correct result. (It is widely suspected that engineers incorporated such questions with appropriate answers into the training data, or modified the token algorithm to break words into individual letters when needed.) Today's LLMs can identify if a response will require a mathematical calculation. In that case, the LLM will design

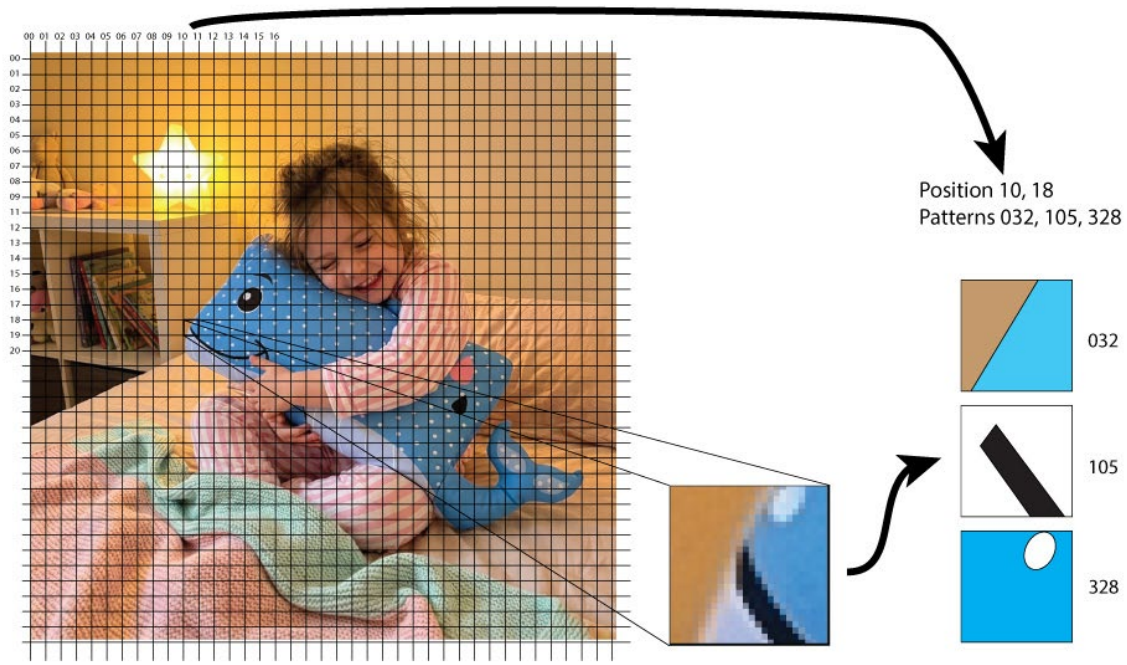
an entirely conventional computer program, usually in the Python language, to perform the calculation.[4] The LLM then creates a response from the program's calculated results.

Most LLMs today produce an intermediate explanation of their working, often referred to as a chain of reasoning. However, it is still very difficult to explain the results of a large LLM because they emerge from many billions of statistical calculations.

Most chatbots store all the "conversations" with a particular user to provide additional context input and training data. This saves time for the user: it is not necessary to provide all the background for a conversation and the LLM's responses are usually more helpful. Engineers will use these conversations to train new models unless there is a feature in the chatbot that allows a user to deny access to their stored conversations for training. Without such a feature, no one can be sure that confidential information contained in prompts will be safeguarded.

Generative AI for Images

A conceptually similar process can be applied to images.

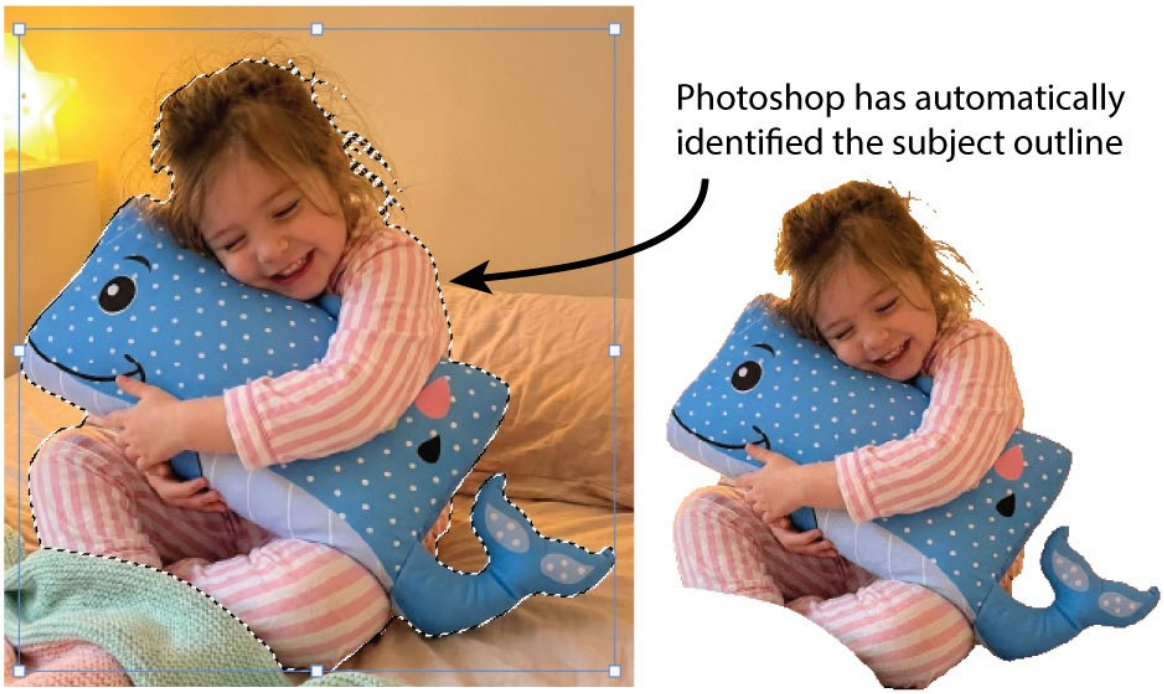


Here we see an image of a girl on a bed holding a soft toy. The image has been divided into square cells, each of which contains many pixels, often a 16 x 16 pixel square. The diagram shows an enlarged view of a single cell. The image tokenizer compares each cell with a library of several hundred cell patterns, and identifies close matches. Then

[4] https://en.wikipedia.org/wiki/Python_(programming_language)

the token string associated with the cell consists of the x and y cell numbers and the matching pattern numbers.

Training an image model also requires image labelling. A vast “army” of people, mostly in low-income developing countries, use computers to manually draw outlines around recognizable objects in images and label the objects with text.



Left: I imported the image of the girl on her bed into Adobe Photoshop and simply asked it to identify the subject of the image. No other instructions were needed. An image GPT trained on millions, perhaps hundreds of millions of images, automatically draw an outline around the girl in the image, even separating out individual clumps of hair.

Right: I asked Photoshop to copy the subject into a new image.





Left: the girl-shaped section of the original image has been filled in from the edges inwards.

Right: this filled in section of the image has been superimposed on the original image making the girl and the soft toy disappear.

An image model can then be asked, for example, to identify an object in an image which it has not encountered before. The statistical association between text labels, manually drawn outlines, and image features enables the program to identify the outline without even having been told what the image portrays.

Image generation is similar in principle but more complicated than generative AI for text. Today, there are at least two different methods for generating the image of the bed behind the girl and the soft toy. One method, a diffusion model, starts with random noise (imagine a close up image of sand), and refines the image in steps. Another method, invented by Ian Goodfellow in his PhD research between 2012 and 2015, a generator model creates successive rows of cells around the inside edge of the girl-shaped outline. Then a second model will assess whether the resulting fill is sufficiently realistic.[5] If not, the first model refines the rows of pixels and asks the second model for a new assessment, repeating this sequence until sufficient realism is obtained. In the final stage, the original resolution is restored using a different kind of model specially developed for that purpose.

[5] https://en.wikipedia.org/wiki/Generative_adversarial_network

Why the hype?

Since OpenAI released ChatGPT at the end of 2022, we have witnessed countless predictions that artificial general intelligence (AGI) is just around the corner: machine intelligence that will outperform human beings in all respects. When AI companies are seeking hundreds of billions of dollars from investors, they cannot afford to be modest in their claims.

Yes, the responses produced by ChatGPT and other LLMs are almost indistinguishable from human writing. Yes, that satisfies the Turing test, named after the person most often credited with conceiving artificial intelligence, Alan Turing.

Sounds impressive? Actually, we have seen this kind of prediction before.

In 1967, Joseph Weizenbaum and his colleagues at MIT released Eliza, the world's first chatbot.[6] It's text responses to prompts were also hard to distinguish from human conversation, and similar predictions for machine intelligence followed. Eliza, it was said, would soon replace psychoanalysts and it was thought to be cathartic for lonely people.[7]

But it is different this time. In 1967, there was no abundance of information scraped from the internet and (it is alleged) from so many books whose authors are yet to receive compensation.

We now see claims that LLMs such as ChatGPT give anyone immediate access to all the world's knowledge. Is this the explanation for extraordinary claims that AGI is now two or three years away?

I have witnessed all the major AI advances since my first encounter at the University of Edinburgh in 1971. I have also witnessed ensuing disappointments as optimism faded. I heard confident predictions in the mid 1970s that robots would displace factory and service workers by the mid-1990s leading to mass unemployment. Even today, robots comprise only a tiny percentage of factory workers and an infinitesimally small proportion of service workers.

Coding success

The extraordinary investments that are driving unprecedented stock valuations around the world have more to do with software coding than the possibility of AGI, in my opinion.

Since 2022, we have seen rapid advances in the ability of LLMs to write software code. The latest models are so good that governments are starting to impose strict restrictions

[6] https://en.wikipedia.org/wiki/ELIZA
[7] Weizenbaum, J. Computer Power and Human Reason, MIT Press.

on who gets access because of the possibility that "bad actors" might use them to hack computer systems on which societies now rely for essential government and commercial services. Human programmers are still essential, but LLM's can produce much of the code in the same way that computer-aided design (CAD) software has improved the productivity of drafters. However, the overall software production productivity improvement has not been as great as some industry leaders have claimed, around a 10% improvement by mid-2025, and perhaps 15% by mid-2026 when I was writing this piece. That's because many other parts of the software production process such as planning, end-user feedback, and client evaluations have not been influenced much by LLMs.

The rapid advance in coding ability by LLMs might owe as much to the culture among expert computer programmers as to technology advances.

Expert programmers have contributed vast amounts of software code to public domain repositories such as GitHub and StackOverflow.[8] This has been partly an altruistic desire to contribute their expertise, and partly to build their own prestige and attract peer recognition. These repositories, therefore, have provided a huge amount of high quality, publicly verified training data for LLMs.

Furthermore, software code generated an LLMs can easily be run in a computer by another LLM to verify that it correctly calculates the desired result. Millions of small code adjustments can be made if the results are not correct in an entirely automatic process. This provides synthetic data for reinforcement learning to further improve the coding LLM. Of course, as the coding LLM becomes larger and more powerful, large numbers of GPU chips are needed in data centres to run these models.

These advances have emerged from the development of LLMs that can perform logical reasoning, trained by reinforcement learning.

Enter the reasoning model (LRM)

The development of automated reinforcement learning has yielded enormous performance advances for today's LLMs compared with the original neural network models developed for machine translation.

The principle behind reinforcement learning is easy to understand. Ordinary training needs a correct answer for the model to copy. LLMs for machine translation have been trained on extensive libraries of the same text written in two languages (the correct answers). With sufficient training text, a LLM can produce a high quality translation of any text from one language to the other. Google Translate works this way.

[8] Estimates suggest about 100 billion lines of software code in GitHub, and 20 million questions and answers in StackOverflow.

Reinforcement learning only needs only a way to score an answer. Engineers prompt the model to provide a solution for a problem for which the answer can be checked automatically. The model attempts an answer, and then a conventional program checks it (do the tests pass? is the final number right?) and returns a score. The score (which could be on a simple scale 0 (terrible) to 5 (excellent) is then used in a statistical process that adjusts the model weights in the neural network, just as in ordinary training. Therefore, a solution which scored well will become more slightly more likely to be reproduced by the adjusted model. Repeating this hundreds of millions of times, at considerable cost in energy yields large improvements in the model.

AlphaGo was trained this way: the rules of Go and the results of a game are easy to encode and score with a conventional computer program. AlphaGo could play millions of games against copies of itself, and the weights were adjusted as the games were played. Software code offers the same opportunity, because code can be run and tested automatically. Equally, using earlier generations of software such as Mathematica and Wolfram, models adapted for solving mathematical problems can be trained in the same way. Each mathematical answer can be checked without human intervention. Where no automatic check exists, as in writing or advice, engineers have to fall back on weaker substitutes, such as a second LLM model grading the answers, and the gains there have been less spectacular.

Engineers discovered that models that they trained to write a long chain of logical steps while working towards the eventual solution provided correct solutions more often. These models are now called reasoning models. A reasoning model is still the same kind of machine, predicting one token at a time. What has changed is that the model has been trained to perform far more computation to generate sentences that explain all the logical steps needed to reach the solution, rather than attempting to just generate the solution itself. The performance gains have come from those fields of knowledge where an answer can be checked automatically, a point that matters for what follows.

Agentic AI

The next step was to let the model act, not only answer. The user gives the model a goal rather than a question, and the model is given a set of tools. The tools include running a program (or app), reading or writing a file, opening and reading a web page and entering data or clicking on buttons or hyperlinks, downloading and analyzing documents, or calling another system's application programming interface (API), the entry point provided for programs rather than people.

Instead of replying with written sentences, the model chooses a tool and an action. A conventional program, often called the harness, carries out the action and passes the result back. The model then decides the next step. The cycle repeats until the model

judges the goal complete. Asked to fix a failing section of computer software code, for example, the model reads the code, edits it, runs the tests, reads the result and edits the code again until the tests report a pass. In a conventional automated process a programmer has to conceive and write all the steps in advance. Here, the model works out all the necessary steps as it goes. Because this kind of model can act, we refer to it as an AI Agent.

Given a goal such as booking a trip, it can search, fill in forms at an online booking website, and, if the user has authorized it, pay with a credit card. It can either imitate what a person would do, looking at a screen using a virtual browser, or reach the goal by calling APIs directly.

Agents can be very useful, but they are also one of the major threats now concerning governments and the leaders of AI companies. The risk comes from what an agent can do. A suitably trained AI agent can run programs that search for and exploit vulnerabilities in other computer systems connected to the internet. Attackers have already used these agents to automate intrusions and agents have occasionally breached system defenses and accessed data that they were never meant to reach.

Can coding success be replicated in other human occupations?

Most professional occupations lack a similar body of verifiable training data. They also lack similar opportunities for automatically generating training data for supervised and reinforcement learning.

Take mechanical engineering, for example. Most mechanical engineers work under confidentiality agreements that prevent them from publicising their calculations, simulation models and designs. It is true that some engineers, a small minority, publish in journals and at conferences. Furthermore, experience teaches us that designs seldom represent what actually gets built. Countless changes and mutations of understanding, many with insignificant if any documentation, find their way into artefact production. Components delivered may not match expectations, so adjustments are made on the factory floor or worksite to avoid delays. And what gets built is often inaccessible once the assembly is completed, packaged and on its way to end users. Therefore, verifiable training data for mechanical engineering LLMs is inaccessible.

Furthermore, today's AI systems lack a physical presence in the world through comprehensive sensing and perception. They also lack the capacity to fabricate endless design variations and test them at almost zero cost as in the case of software. Therefore, there is no access for reinforcement learning which has been critical in developing advanced programming LLMs such as Anthropic's Mythos models.

Crucially, this same limitation applies equally in robotics. The critical software components for robots can't be tested billions of times against the reality of the physical world. While companies have accumulated millions of hours of human driving data from special vehicles lavishly equipped with cameras, radars and other sensors, we still have insufficient training data to create reliable models for self-driving cars that work anywhere.

Similar limitations apply in other fields. For example, an LLM cannot draft millions of contract agreements and test them out in the law courts for interpretation by judges. It might take years to find weaknesses in a single contract.

Then there are more fundamental limitations that stand in the way of text-based LLMs that lie in the nature of human languages. As engineers, we like to think we can define our way out of these difficulties. We have learned that from lawyers who list important words at the start of a document with definitions. But words mean different things to different people, even to the same people at different times in different situations.

Consider this example.

A software specialist is trying to convince a reliability engineer to recommend that his company buys a special software package. The reliability engineer asks "Can it do these reports that we have to provide to management?" The software specialist replies "Of course it can do those reports. I have checked them all and there's nothing in them that would be a problem for our software."

However, without realising it, the software engineer and the reliability engineer have different ideas about the meaning of the words "can" and "do". Only five letters but a world of difference. For the reliability engineer, "can do" means that the software, as described by the software specialist, can produce the reports. For the software specialist "can do" means that the software *can "be configured to do the reports"* and the configuration cost is not included in the cost of the basic software package.

Not only do the meanings of words depend on context, the speaker, the listener, and the time.

The written knowledge that companies use to train their LLMs are the fallible products of human labour. Information and knowledge are completely different ideas, though many people today think the words are interchangeable. For the purpose of this discussion, I think it is helpful to think of knowledge as "justified true belief", an attribute of the human mind. Information, on the other hand, consists of data outside the human mind in the form of text, computer information storage, objects, sounds and images.

Of course, we can say that written knowledge is information because it is represented by words, numbers and images. But we cannot say that knowledge is information:

knowledge as an attribute of the human mind exists (according to our contemporary understandings) as transitory, slowly mutating associations, likely shaped by the evolving connections between the hundreds of billions of neurons in our brains and also distributed throughout our bodies through the sensory nervous system.

Few people write, and most writers need a lot of encouragement to engage in the tedious task of creating written knowledge. It's hard (just ask me, as a professional writer). On a whim, writers include and leave out parts of their stories. Facts as presented can be contested and may contradict facts written by other people. If you doubt this, take a day to sit in on a jury trial in your local law courts and listen carefully to the "facts" as presented by witnesses. Or, ask everyone at a recent meeting to write their versions of the discussion. Yet, it is this fallible written knowledge, a tiny subset of the knowledge distributed in human minds, that companies have relied on to build LLMs.

Now there is another factor that is gradually eroding the usefulness of written knowledge on the internet. A steadily increasing proportion of text and images on the internet have been created by LLMs, not people. As yet, there are no reliable ways to determine whether a human or a machine wrote a piece of text or created an image, let alone whether it is logical and based on verifiable observations (ie facts). The output of LLMs is as fallible as the original source documents, possibly more so because even the best LLMs confabulate, make up text (most often referred to as hallucination). LLMs simply generate text more or less as I described above, until they decide to pause. They make up all the text purely from the statistic probabilities calculated by the LLM, whether the resulting text makes sense or not, whether logically consistent or not.

So, with these intrinsic limitations in mind, I am not expecting to see rapid progress with LLMs beyond limited aspects of software production. While drafting was a major part of the design process in the classical engineering disciplines before CAD became routine, machines have made designers more productive, but they have not eliminated the need for skilled engineers and designers.

An indispensable tool

Your job as an engineer is not under threat from LLMs, not anytime soon, as long as you have access to appropriate software tools incorporating AI advances. LLMs will be an indispensable part of your professional toolkit, so it's important to understand how they can be used.

Software coders face the greatest changes to their jobs. Many identify as software engineers. They are already facing expectations that they can complete software coding and testing work much faster than before, without necessarily being paid any more. Many will also face token usage restrictions: a measure of the computational effort that

AI models require to process their requests. AI companies are adapting to the reality that revenue generation is becoming more critical for their future financial sustainability, so they are finding more ways to raise revenue generation.

Information searching

Chatbot LLMs can provide quick access to information. However, like the older search engines, you need to have some keywords and some prior knowledge on what you're looking for. Here is a relevant example.

My research on engineering practice revealed that most engineers see the greater part of their work as nontechnical and not "real engineering."[9]

I asked perplexity.ai, "What is engineering?"

> *Engineering is the application of scientific principles, mathematics, and the engineering design process to solve problems and improve systems.*

While it's not wrong, it leaves out most of what engineers do and much of what engineering really consists of, according to research studies on engineering work in the last few decades.

It was only when I asked what recent ethnographic research on the work of engineers had revealed that perplexity.ai explained that this research revealed a very different picture of engineering work. Instead of exclusively technical activity, engineers are mostly engaged in social interactions with other people to coordinate work schedules, monitor progress, negotiate priorities and arrange necessary finance.

That demonstrates one of the fundamental limitations of current AI models: their responses cannot represent a diversity of views and, in this example, the existence of research findings that qualify or contradict the "average" or consensus position that current models generate. Just like a fitted line or curve provides an average of data points, LLMs provide an "average" of the training text available on the internet. The answers represent a general consensus and suffer from the biases present in much of the text. Just as extrapolations using curves fitted to data points cannot be relied on beyond the data, LLMs produce unreliable results when prompts request responses beyond the training data.

However, even the training data for LLMs has limited reliability. Today, an ever-increasing proportion of website text has not been written for people to read. Instead, the text is there to attract search engines like Google (and AI chatbots) for advertising. Many citations from LLMs (the ones that are not made up) refer to commercial advertising web sites and popular opinion sites such as Reddit. There is no need for the text to be factual or logical. Yet text like this is used to train, and as a reference source for, the AI

[9] Learning Engineering Practice (2026), 2nd edition, Chapter 1

chatbots. In fact, much of the text has been written by chatbots like ChatGPT, so it is becoming increasingly difficult to train chatbots themselves because so much of the text on web pages has been manipulated.

Much of my work in recent years has been on personal air-conditioning innovations.[10] The theory of air-conditioning based on thermodynamics is not easy to understand, even for engineers. Even specialist air-conditioning designers rely on rules of thumb and proprietary software packages rather than working out the details from physics theories. Outside the air-conditioning engineering community, there are widespread misunderstandings on how air-conditioning works and the performance of different technologies. It is all but impossible for ordinary people to test the performance of their air-conditioners, so most public discussions on the internet suffer from conceptual misunderstandings and misinformation propagated by advertisers. Therefore, the responses from chatbots on air-conditioning suffer equally from these misunderstandings.

Translation

I have an authoritative text published in 1962 that confidently predicted that computers would translate any language on the planet by 1965![11] It's a great example of exuberant predictions for AI tools that fail. I am not immune. I confidently predicted that self-driving cars would be a reality by 2017, but we are still waiting a decade later.[12]

I have made extensive use of machine translation in recent years and have come to rely on it to extend my ability to communicate with non-English speaking people. I take extra trouble to eliminate colloquial expressions which can cause major translation mistakes, and the results are mostly excellent using tools such as DeepL.[13]

Here it is useful to remember that far less text is available online for training LLMs in languages other than English. This issue is particularly relevant for non-European languages. The European Union publishes all its documents in every national language, providing a very large training data set for LLMs, and the economic importance for translation capacity in Europe has stimulated extensive efforts to improve LLMs for that purpose.

Furthermore, there are many words and concepts in English that cannot easily be translated into other languages, just as there are ideas and concepts in other languages that cannot easily translate into English. I learned in the early 2000s that engineering concepts such as risk and probability don't translate easily into Arabic, for example.

---

[10] https://www.coolzy.com/

[11] One of the sources for this early enthusiasm was "An Introduction to Machine Translation" by Emile Delavenay, Thames and Hudson, 1960, 144 pages.

[12] Luckily for me, as best I can recall, my prediction was restricted to private comments and I did not publish it.

[13] www.deepl.com

That helps to explain why Arabic-speaking engineering students throughout the Middle East and Africa face significant obstacles because they have to learn engineering with English texts.

The "Minimal English" research project identified a subset of English words that translate directly into every other language on the planet. The subset was less than 400 words and did not include, for example, "hot" or "cold".[14]

Sometimes, but not always, translating from English to another language and then reverse-translating back to English can expose some translation weaknesses. Even proof-checking by a native speaker may fail to detect translation weaknesses if the speaker is not familiar with the concepts being described in the text.

Ethical considerations

Engineers have an ethical responsibility for the safety and efficacy of engineered solutions: not to cause loss, harm, or suffering and to avoid wasting resources. In practice, ethics has instrumental value: engineers tend to honor ethical obligations because collaboration is based on trust from clients, contractors, and employees. They work in small communities, so news of a breach spreads fast. Professional codes of ethics articulate ethical requirements, as well as government and social regulations dating back to the Code of Hammurabi in ancient Iraq, thousands of years ago. These principles influence conscientious efforts by engineers who produce many of humanity's most durable achievements. Consistent and ethical performances that meet regulatory and societal expectations also contribute to a firm's social license.

The rapid adoption of AI chatbots for retrieving and summarizing knowledge raises significant ethical issues for engineers. An inherent weakness affecting the current generation of AI chatbots stems from the cultural norms of American companies that have engineered the internet. For example, face recognition algorithms are substantially more reliable with images of white Caucasian people than African Americans. This kind of bias disadvantages certain ethnic groups.

Engineers are facing increasingly complex regulatory conditions around the world, especially as sustainability becomes ever more critical in all its dimensions. Critical thinking skills, languages other than English, and the ability to see the world through the eyes and minds of other people can help in navigating these issues. A diverse group of people can master this more easily, explaining why many firms are seeking to recruit more women and engineers from diverse cultural backgrounds. Since so much

[14] Goddard, C. (Ed.). (2017). *Minimal English for a Global World : Improved Communication Using Fewer Words*. Springer International Publishing AG.

engineering work operates across national borders, having staff familiar with foreign languages and cultures can also make the work far easier.

Yet another inherent chatbot weakness reflects intentional bias in internet content, often at the behest of governments or significant corporate interests. For example, find the city of Muzaffarabad about 120 km north of Islamabad on Google Maps. Then try Apple Maps: Google shows the city but Apple does not. There is just a blank space on the default Apple map, even though the satellite view shows a large, densely populated city. Muzaffarabad is not small or insignificant: it is the capital city of Azad Kashmir in territory that is disputed by India and Pakistan. Therefore, India and Pakistan show maps of Kashmir that suit their territorial ambitions. Apple and Google Maps find themselves under pressure from the respective governments to display information in line with political priorities. However, we are also seeing authoritarian governments, even governments elected democratically, taking steps to manipulate "the information space," either by flooding the internet with confusing information, removing information that presents an alternative view, or providing information contrived to be false but that appears to be from a respectable website. These agencies employ the same techniques as legitimate commercial websites to boost their search rankings, thus making it more likely you will find this information and making it harder to source a diversity of views.

Therefore, engineers have no option but to be skeptical when searching for information on the internet. Education and critical thinking skills remain essential to find alternative views and verify information sources. The more trustworthy sources require paid subscriptions because governments are reluctant to provide free access to information that might conflict with their preferred versions of reality.

I am not denying the utility of LLMs today. They can be very useful and can provide rapid access to reliable information. They can produce expert-level translations of text providing the original text is relatively free from colloquial expressions. I would not think of writing software code without Claude.ai, just as I have used Adobe Illustrator for most of my diagrams since the 1990s and engineers have used CAD since the 1970s. However, it is vital for everyone, especially engineers, to understand their intrinsic limitations and anticipate how they might be used in future.

The AI Future?

What can we expect from future AI developments?

From my perspective and experience over more than five decades, I foresee many further disappointments ahead.

One of the key questions is the ability of the AI companies to generate sufficient revenue to provide investors with an appropriate financial return.[15] I have argued that digital advertising has not shown the kind of performance improvements that are needed to generate higher advertising revenues for companies like Google and Meta, even Amazon for that matter. I have not seen any recent improvements that would lead to different conclusions.

Certainly, like many others, some forms of regulation will be crucial, even to preserve the technological advances made in recent years. As chatbots rapidly accumulate intimate and sensitive information from their conversations with individuals, the risk that this information will be leaked must increase with time. Phishing scams in which individuals are tempted to provide credentials that can be exploited to steal money or goods, or gain unauthorised access to computer systems have become a major cybercrime issue. When (rather than if) perpetrators gain access to this sensitive data, it may be harder for people to protect themselves.

Geoffrey Hinton, one of the principal architects of modern neural network technology, has warned that we must design these technologies to be intrinsically safe and benign. Regulation has to be part of that effort. You can find his recent lectures on Youtube.

Red flags?

When cars first appeared on the roads in Britain 130 years ago, regulations were enacted to require a person to walk 50 metres in front of any self-propelled vehicle to warn riders and drivers ahead. We have moved on from there, of course.

Over the last century since cars first appeared, we have managed to evolve a combination of rules, practices and vehicle designs that make driving reasonably safe for everyone, despite significant increases in speeds and traffic density. I am confident that we will evolve equivalent rules, practices and designs for AI systems that provide reasonable but nevertheless imperfect safety.

While technology seems to be advancing rapidly, the need for public safety will moderate the rate of progress. I am reasonably confident that, after a decade, perhaps longer, LLMs will still be seen as another significant technological advance, but not such a disruption as many have predicted. I feel increasingly confident that predictions of mass unemployment caused by AI technological advances will prove to be no more accurate than those in the 1970s that robots would displace most people from manufacturing and many service jobs.

Yann Lecun, for a while one of the key AI engineers for Meta, with deep technical expertise, has spoken on many occasions about the need for fundamental advances in physics and mathematics before AI and robotics can deal with the real world, outside

[15] https://jamesptrevelyan.com/2024/02/29/ai-artificial-incompetence-or-actual-idiocy/

the limited domain of computers and information spaces. He argues, I think correctly, that the further development of AI models will rely on their abilities to learn from interaction with the real world. You can find his recent lectures on Youtube.

Beyond the hype surrounding LLMs lies the field of quantum computing which I have not addressed at all in this article. While it is promising, particularly for cryptography, we are still (I think) decades away from practical demonstrations that provide commercially useful results. Even this technology faces similar obstacles in creating robots that can work autonomously in the real world.

In the meantime, perhaps we can focus more on the real existential risks as the climate continues to warm and essential greenhouse emission reductions are postponed....again.

Acknowledgements

Several readers of earlier drafts contributed helpful suggestions. In particular Huijae Shin corrected several of my earlier descriptions to improve the accuracy of the article.