

Contents:

[Natural Intelligence about the Artificial One](#)

[1. What is intelligence?](#)

[2. Will AI be able to do my job better than I?](#)

[3. What can we do there?](#)

[3.0. Starting from the left side: Intelligence](#)

[3.1. Raw Power](#)

[3.1.1. No AI](#)

[3.1.2. "Incomputable" Functions](#)

[4. Data](#)

[4.1. Value\(s\)](#)

[4.2. Examples](#)

[4.3. Features](#)

[4.4. Labels \(and Kalpavriksha story\)](#)

[5. The bottom line](#)

[6. What's next?](#)

Natural Intelligence about the Artificial One

With arrival of AI, many are worried.

Would it eliminate our jobs? Or make them 10x easier? Is there anything that AI cannot do? How should we prepare?

I've been pondering these questions, too, and decided to share my framework for thinking about them. You don't have to adopt it. But it helped me understand some things, so maybe it would help you, too.

Most of this is **not** based on the most recent AI news. It is, rather, a product of a decade-old reflection. But it is certainly the most recent advances in the AI that have prompted me to write the things down.

With that, first things first:

1. What is intelligence?

The truth is, we don't know. Only Wikipedia lists about 10 definitions^[1], and there easily could be more. How can we reason then about "artificial" something if we have no idea what a natural one is?

Fortunately, to address the questions above, we don't really need to know. What we need is what could be called a Projected, or a Measurable Intelligence. It is simple:

- 1. Pick a functionality that is at least somewhat related to intelligence. Say, numbers addition.
- 2. Generate for it multiple tests with known correct answers. E.g., "1+1=2", "7+11=18", "0+7=7".
- 3. Run them through a system in question.
- 4. Measure the % of the correct responses.

Yes, for some scenarios this is not so easily applied (we'll touch them later). And yes, this approach measures only a slice, a narrow projection of what a "real intelligence" is. Still, it provides us with two important things:

- Connection to practice. For most of everyday life it does not matter whether an instance of AI is "intelligent". It matters if it can drive a car, or write code, or tell cats from dogs with sufficient precision. If you are concerned about AI taking your job, or making it better, Projected Intelligence carries the answer.
- Multiple projectability. A system can be tested against a virtually endless set of "simple" independent projections, from text understanding to an IQ test to discerning emotions or to drawing funny cartoons. If it does no worse than a human on **all** of them, it would be hard to advocate that the system is not at least as intelligent as a human is. The reason, while statistical in nature, is intuitively simple. If something consistently projects large shadows regardless of the direction of light, it is very unlikely to be a thin pile of garbage carefully arranged to look like a human being ([2]).

So, while it is conceivable that alternative or "hidden" types of intelligence can possibly exist, pretty much everything we do that someone cares about is measurable, even if by a subjective yardstick. As a result, a sufficiently diverse set of Projected Intelligence tests can probably get us arbitrarily close to measuring a "true" intelligence of a given system.

Therefore, if we care about the questions that open this article, we can use the Projected Intelligence in place of "intelligence", at least within this discussion.

2. Will AI be able to do my job better than I?

The short answer is "Yes, eventually" (though not necessarily "soon") for most of the intellectual jobs. And this, actually, is neither bad nor news.

As long as we measure AI performance against a specific job, we subject it to a Projected Intelligence test. Structurally, such test could be represented as a Test phase of a classic ML supervised learning:

Test	Expected	System Response	Correct?
1+1	2	2	Yes
7+11	18	8	No
0+7	7	7	Yes
...

In order to be able to do some job, an AI needs to pass the above test within that job's domain. But that test is merely an approximation problem: given an unknown function $F(\text{inputs}) \rightarrow \text{outputs}$, and many (possibly implicit) input/output training examples, an approximator needs to learn F and prove that by correctly computing F on the inputs not seen during training. The more complex structures of F an AI can handle, the more advanced intellectual jobs it can take.

And here is the important point. In 1956-1957, Kholmogorov and Arnold have proven^[3] that any real-valued multidimensional function could be represented as a *finite* composition of simpler (in fact, single-argument) function superpositions and additions.

In a simplified language that means that (almost) any complex problem could be reduced to a finite sequence of problems as simple as arithmetic. Thus, an arbitrarily complex intelligence could be exchanged for a finite amount of raw CPU power.

Of course, in 1950s this knowledge was very far from usable. The theorem did not specify *how* to achieve such a representation, and even in cases when that was clear, the "finite" was very much like "infinite" in most respects. And there are limits to the applicability of that theorem, some significant and important (we'll talk about them later).

But it broke the dam, so to speak. People followed with a multitude of improvements, and by ~2000 it was pretty clear^[4] that the answer to the problem of "learn to approximate (virtually) any problem" was a neural network with enough CPU + memory behind it. Some nearly equally capable alternatives^[5] have also been developed.

[Curiously, people trained in Physics have often been aware of that at least intuitively. Physicists have long known that "the reality is decomposable" -- that most complex functions of Nature are approximable as rather small superpositions of a few common functions. Learning Physics involves heavy and extensive training on understanding when such decomposition is justified. Mathematicians who have not been through it are driven nuts each time they hear a Physicist say, "let X approximately be..." or "let's ignore Y..." because they need a formal proof, rather than an intuition, with respect to their cry of "why!?"]

To summarize: in principle, there is no obstacle to an AI doing (almost) any kind of intellectual job better than you or I, at least when measured by tests. Surely, some jobs will come earlier, and for some it would still take decades, but there is no fundamental impossibility here. Accepting that as the first order description of a future is helpful. Because it enables one to focus on the actionable question of "what shall I do?" instead of contemplating the less important "if?", "when?", or "why?"

So, let's talk about that. Let's see what we the humans could and should do in the world cohabitated by an intelligence potentially much more powerful than ours.

3. What can we do there?

Sci-fi sometimes speculates (e.g. [6], [7], [8], [9]) of intelligent beings not just smart, but *irreducibly* super-smart relative to humans. Their solutions to practical problems not simply dwarf any of humans' but are fundamentally unintelligible to us due to our biological or physical limitations.

Maybe sadly, such implication appears not necessary. There is no need for a special "other" intelligence. As it seems now, the formula for *any* level of a Projected Intelligence boils down to this:

$$\text{Intelligence} = \{\text{Raw Computing Power}\} \times \{\text{Data}\}$$

And as the {Raw Computing Power} is a fixed parameter for us humans while ever-increasing for AI, it begs for a question: what can we do to remain competitive or at least mutually beneficial to the AI?

Let's break the above "formula" into parts and look at each one closer.

3.0. Starting from the left side: Intelligence

Of course, not everything is Projected Intelligence -- or not easily "Projected", at least.

No, I am not talking about art, or building social connections, or subjective feelings. All these things are measurable, even if by noisy or imperfect tools. If art is "anything that a creator and at least one other person calls art", then the response from that "other person" is a measure or a test. If social connections change how the interconnected group behaves, that change could be measured. Subjective feelings do have an impact on actions or, at least, on the scans of brain activity. Even if cumbersome and expensive, that is still a test.

I am talking about situations where tests with the correct answers are unknown, very limited in number, or are not straightforward to administer.

Evolution. A single gene mutation can cause an organism to die or to become a king of an ecosystem. Sure, we can predict the effects of *some* mutations, and far from all social success is driven by genes. But ultimately the remaining 20% of the test is just... life. Evolution works backwards, calling something a success or a failure only in retrospect, sometimes very late after a change to be tested has been made.

Science. People who dug up dirt near Ytterby^[10] did not have a goal of using Terbium in military sonars 150 years later. They just executed their curiosity and tried to close the gaps in their knowledge. The primary value of science is (usually) not to check off against known labels. It brings value by **expanding the set of data and labels** that could be used to validate or predict **other** results. As an exploratory process, it can be evaluated only by considering the quality of predictions

on all questions known to humanity, with training on all data historically known up to date, and then measuring the delta of that quality after bringing in the new discoveries. Even after serious unavoidable simplifications, this is a very expensive (and usually neglected) process.

In these situations, we humans tend to replace the true correctness with social institutions that assign artificial metrics to approximate it. It usually works, as evidenced by us still being alive and harnessing tremendous volumes of knowledge. That may even cause thinking that science is a 100% social construct, but it is more complex. A binocular is a purely artificial construct, but a picture you see through it is not. Science may look like a social phenomenon, but the results you get with it often are not. Their correctness is evaluated by something ultimately more powerful than even the strongest social opinion, for wrong choices in scientific priorities could easily eliminate us all, possibly with everyone's cheerful approval and aspiration.

If you work in an area like that, not really much is changing for you. To assure success, you still need to talk to *humans*, and talk *a lot*, and constantly assure them that some of your work needs to be evaluated on metrics more complex than the count of projects closed. You can use the AI for that, or to aid with your work directly. You can probably cooperate with it, if it turns out that it can do intelligence beyond the Projected/Measurable one. Regardless of that, being in this area is tough, and it has always been.

And no, I don't know whether the AI can do more than just the Projected Intelligence. It possibly can. I just have no tools to reason about that.

3.1. Raw Power

Let's not completely discard it. There are still scenarios where a human "CPU" remains competitive going forward.

3.1.1. No AI

AI will be used broadly, that's for sure. But would it really be EVER-present and EVERY-where? Probably not. Probably there would be gaps and interruptions and imperfections and situations when you just can't use it. Then, being able to do what an AI does could offer tangible benefits.

GPS is the most obvious analogy. It is nearly ubiquitous, yet navigating your own city shortcuts without constantly referring to a (discharged) phone is still a power.

Or consider calculations. I can approximately compute logarithms and square roots in my mind. I do that about 1,000,000x slower than a computer, and with about 1,000,000x greater error. Yet that skill has saved me months of debugging, by letting detect gross wrongness in computations early enough in work cycle. Also, probably half of my designs have been created while driving. Why? Because I could perform certain emulations "in my mental CPU cache", with no help from the external tools.

Let's also not forget about the ability to speak clearly and beautifully without AI, especially if everyone around needs it. That certainly carries at least some social value :)

So don't throw away your "weak" "CPU" yet. It can probably still augment the external AI in a lot of surprising ways.

3.1.2. "Incomputable" Functions

The Universe is full of problems where having even orders-of-magnitude upper hand in raw computational power provides rather minor advantage in outcomes.

* Systems that expose Dynamic Chaos^[11] behavior. Turbulence^[12], weather^[13], and Lorenz System equations^[14]. Heat Equation^[15] run in reverse. Abel transform^[16]. Dynamical billiards^[17]. And many, many more.

They all are very difficult to approximate with simple functions, because their solutions could be very (or even infinitely) sensitive to small variations in the initial conditions. An imprecision at the start, no matter how minor, can grow with time, until (rather quickly) the prediction has nothing to do with the observations. ML can still learn and predict these systems to a certain level of detail^[18], but increasing quality of that prediction linearly (in terms of time horizon or precision) may require *exponentially* more computation resources^[19]. This has nothing to do with numeric instabilities or with a choice of an algorithm. This is how these things behave in a physical world.

If a problem at hands has elements of Dynamic Chaos in it, even a weak "CPU" could produce only marginally weaker results compared to its advanced counterpart -- and thus could meaningfully compete or contribute to them.

* Problems where the exact solution has a very high complexity cost -- like a Traveling Salesman^[20] ($O(N!)$) or building the best schedule out of jobs with possible interruptions^[21] (which is $O(2^N)$). Of course, in everyday life they are solved with faster heuristics. But heuristics are approximate, their space is vast, and they may depend greatly on the details of problem formulation or assumptions made. That means that even relatively weak approximators can sometimes gain an upper hand in this space.

As an example, consider a prioritization problem. You have N activities to complete. What should be the starting order for each of them that maximizes some outcome (e.g., time saved after all items are finished)?

In the most general form this problem is difficult because the items in the list can have arbitrary, complex, non-additive interdependencies or mutual restrictions. E.g. what if completing any single item changes the complexity & cost of all other items? Or what if, depending on whether A or B has finished first, the very need to do half of the remaining list changes? The set of possible types of such constrains is likely infinite. That means that the only algorithm that **guarantees** finding the best execution order is an exhaustive search, which considers each of the possible schedules, and evaluates the likely outcome for each of them. The cost of such an algorithm would be $O(N!)$.

Even for 20 items, $N!$ is practically non-enumerable as it evaluated to $\sim 2 \cdot 10^{18}$ combinations. Yet the ideas and TODO considerations of any living person run in 100s [I suggest you try that. Open a text editor and start writing down ANY things you consider necessary, important, or interesting to do. If you spend at least couple days on that, you'd find yourself in a 3-digits realm.]

Of course, nobody solves this problem via an $N!$ -way. We use heuristics ("start with things that have the most chance of being crossed off the list today"). We apply simplifications ("most items are approximately independent of each other"). We group and re-group work items (with "fix the kitchen sink" and "buy new light fixtures" getting lumped into "home improvements" project). Each approach to simplifying and representing a problem is a heuristic, almost guaranteed to be unique, and, when competing with other heuristics, not unlikely to be the best at least on **some** data. The heuristics set you apply to scheduling and prioritization of your life **is** your identity. It is **what** you (eventually) are. If you don't have a schedule of your own that you hone and protect, you don't have an identity.

Let's run an experiment:

Dialog #1:

Me:

Imagine you are a parent of a 3 year old girl. She needs to do four things (listed here in no particular order):

** Change a band-aid on yesterday's scratch on a knee.*

** Eat a candy*

** Take a bath*

** Brush her teeth*

What should be the best order of doing all those four things? Think like a parent living in a real, physical world. Only state your response and the justification for it, do not re-state the problem.

LLM:

The best order to do these tasks would be:

Change a band-aid on a yesterday's scratch on a knee.

Take a bath.

Eat a candy.

Brush her teeth.

Justification:

First, you should change the band-aid on the knee. This is to ensure that the wound is clean and protected before the bath.

Next, the child should take a bath. Bathing after changing the band-aid ensures that the band-aid stays in place and the wound stays clean.

After the bath, the child can eat a candy as a reward for behaving well during the first two tasks.

Finally, the child should brush her teeth. This should be done last, especially after eating a candy, to maintain good oral hygiene and prevent tooth decay. It's also a good habit to brush teeth before bedtime.

(The end of Dialog #1)

This list has only 4 items, so there are only $4! = 24$ possible orders of arrangement for them. It seems that on such a small space of solutions the best one should be easily found and unique. Yet *I* would argue that we should first take a bath, and then change the band-aid. Why? Because a wet band-aid comes off easier, so it would be less painful to change it, and a new one would more likely stay put if applied after bathing, not before it.

Does that mean the LLM was wrong? Not at all. Its reasoning was clear and quite solid. My plan was probably better, but the LLM also produced a usable plan.

So, while both being nearly right, we diverged on a plan with only 24 possible permutations. What would happen if their count reaches 10^{18} ? There is no way we would agree.

Miraculously, juggling a list of just ~ 20 priorities makes (virtually) anybody effectively unpredictable to the most advanced forms of ML.

While possibly reassuring, that can also trigger a downward spiral of capitalizing on inferior solutions.

How?

Suppose you can solve some problem with 100% correctness. And someone else, capable of fully predicting that rational solution of yours, is already waiting at your point of arrival to snatch all the benefits from you. Your chance of gaining in this game is 0%, then.

A perfect solution could be spoiled in virtually infinite number of ways. So, you can opt for one of them that results in only a 60% chance of success, and a 30% chance of being interdicted by someone else. Then, your chances of gaining something in this game are suddenly $60\% * (1 - 30\%) = 42\%$. By being unpredictable, even at the cost of accepting an inferior solution, you gain. That's how being deliberately irrational could be a strategy for someone competing against a more advanced mind. At the very extreme, falling back to pure randomness (as modeled by Ph. K. Dick in "Solar Lottery" mental experiment [\[22\]](#)) completely levels out the chances against any adversary.

[Some text was replaced with its SHA256 value. Reason: "sensitive topic", Hash:5D-BD-39-9C-1C-87-61-82-AB-25-0F-FC-E0-03-37-EA-62-33-19-25-E5-22-2B-CD-D8-14-1C-09-7A-E9-73-79]

But why do we think "adversarial"? I do hope actually that our interactions with the AI would be predominantly cooperative. But, in order to effectively cooperate, one must be able to bring something to the table that the other party needs and cannot obtain very easily. Thus, an ability to cooperate implies an ability to compete, and that is why we need to consider that scenario, at least.

The rest of what LLMs would seem unable to do are even "harder" things:

- * Systems fundamentally unpredictable, like quantum noise.
- * Problems formally undecidable (like Halting Problem^[23]) or intentionally designed to require incredible complexity (like Ackerman function^[24] and one-way "trapdoor" functions).
- * Cryptography, if done right.

With that, let's move further.

4. Data

Remember, our "formula" for Projected Intelligence is:

$$\text{Intelligence} = \{\text{Raw Computing Power}\} \times \{\text{Data}\}$$

We've looked at the left side, and at the Raw Power. Now, what about Data?

Logically, any approximation problem could be represented via the same standard structure:

Example	Feature1	...	FeatureN	Label
Ex0	value01	...	value0N	label0
Ex1	value11	...	value1N	label1
...
ExN	valueN1	...	valueNN	labelN

While the AI may not use exactly that representation internally, it is still a good way to de-compose the "Data" into elements and consider the role of each separately.

And we will start with the smallest one: value.

4.1. Value(s)

A value is a single data point within that table, sitting somewhere at the intersection of an Example and a Feature.

Certainly, possessing a unique knowledge about some cell value may offer an advantage... but a very limited one.

That's because almost any data is a label in someone else's model. If strongly needed, it can often be re-created or approximated from alternative data.

Even when not, good decisions are robust against small fractions of missing data. They try not to rely upon few "aha!" values. Therefore, the AI may not be in a high need of singular "priceless" data points.

That's said, there is one potentially useful exception to that. Some values can neither be computed nor circumvented, by design. I am talking about encrypted data and the secrets keying it.

Regardless of whether AES survives the AI, *some* cryptography will be in use. Any handshake within it would rely upon shared secrets that are unique and incomputable. Therefore, generating and remembering strong unique secrets would remain an effective tool for asserting one's identity. And the most natural way to produce such secrets is from "values" that you, and only you, possess.

How could that look? Say, as a child I've spent countless days in the backyard of my grandma's house. I remember it vividly in numerous details. Today, that house does not exist. Other than myself, only two people in the world remember it, and definitely not as clearly as I do. By moving through the mental picture of that place and naming the objects there in accordance to certain rules I can generate long, complex, unique, hard-to-compute but easy-to-remember passwords.

This becomes particularly relevant in the world where AI can easily deepfake anyone's appearance or body language or voice, and where harvesting petabytes of semi-public (and often hackable) data allows inferencing the answers to common identity verification questions -- like someone's car model of 2007, or whether they owned BA back then, or who was their childhood best friend. Ironically, strong personal secrets -- and thus passwords -- would seem to be making a comeback, to a degree.

But other than that, remembering isolated data points -- such as the exact position of "<=>" within C++ operators precedence order, or solubility of GeS_2 -- would probably be of a little use. [Some text was replaced with its SHA256 value. Reason: "Unpopular opinion", Hash:EB-67-04-EE-C7-6C-7F-46-F8-14-97-39-61-43-D6-C9-D8-55-D8-EE-B7-C3-39-46-B4-26-B0-5F-69-09-41-75]

So, let's move to a potentially more powerful data class:

4.2. Examples

Examples are **rows** of data in the logical table above. They are more than just values. They represent knowledge projected upon multiple dimensions. They are what we call "an expertise".

You possess one if you know enough about 2000 elections to discuss their social, economic, foreign relationships, and other impacts. Or if you remember multiple facts about Pluto such as its mass and size, details about its atmosphere, geology, history of exploration, and can place Pluto within the context of other planetary bodies.

This suggest a way of challenging LLMs based on extensive, deep experience that is practically important but that the AI has not had a chance to learn yet. Like, coding in an uncommon programming language. Or deep personal experience with customs, culture, and inner dynamics of some tribe. Or having spent 20 years studying trifluorophosphine complexes of transition metals. LLMs do rather poorly where valid training examples are sparse and scarce, so someone possessing a good bag of them can beat LLMs 10:1 there.

Still, this is a defeatist path. Earlier or later, LLMs will encounter the John F. Nixon's book and comprehend it, **all at once**. Or they may master your area from chatting with other experts. Betting on keeping an important, information-rich area secret for a long time is risky; not using the LLMs power to work with it is, eventually, counter-productive. It may be OK only as a short-term strategy.

Is there a better one?

You can use your expertise to prime the AI via prompt engineering to solve problems relevant to you. A technique called Few Shot Prompting^[25] seems to be close to that. The idea is simple: in addition to asking an LLM a question, you also offer it a few hints or examples of correct answers.

The quality of those examples, directly based on your expertise, can boost the quality of the responses.

Consider this dialog:

Dialog #2:

Me:

The following C# code does not have any errors and does exactly what the author wanted it to do. What does it do?

```
int n = 100;
for (int i = 0; i < n; n++)
{
    Console.WriteLine("*");
}
```

LLM:

The provided C# code seems to have a logical error. The loop is supposed to iterate n times, but instead of incrementing the loop variable i , it increments n . This results in an infinite loop because i will always be less than n .

Here's the corrected version of the code:

C#

```
int n = 100;
for (int i = 0; i < n; i++)
{
    Console.WriteLine("*");
}
```

This corrected code will print the asterisk () character 100 times. Each asterisk will be printed on a new line. If the original code was indeed the author's intention, then it*

would result in an infinite number of asterisks being printed, which could cause the program to become unresponsive.

(The end of Dialog #2)

So, the answer is wrong. Now let's augment the question with a bit of extra expertise (in bold):

Dialog #3:

Me:

The following C# code does not have any errors and does exactly what the author wanted it to do. **Keep in mind that all integer types in C# can overflow.** What does this code do?

```
int n = 100;
for (int i = 0; i < n; n++)
{
    Console.WriteLine("*");
}
```

LLM:

The provided C# code is a loop that will print an asterisk (*) on each iteration. However, instead of incrementing the loop variable *i*, it increments *n*.

Given that integer types in C# can overflow, this means that *n* will keep increasing until it reaches the maximum limit of an integer in C#. Once it exceeds that limit, it will overflow and become a large negative number. At that point, *i* (which is 0) will be less than *n*, causing the loop to terminate.

So, this code will print an asterisk (*) a very large number of times (more than 2 billion times, which is approximately the maximum value of an integer in C#), then it will stop due to the overflow of *n*. This could cause the program to become unresponsive due to the large number of iterations. Please be careful when running such code, as it could consume a lot of system resources.

Here's the corrected version of the code if you want the loop to run exactly *n* times:

C#

```
int n = 100;
for (int i = 0; i < n; i++)
{
    Console.WriteLine("*");
}
```

This corrected code will print the asterisk () character 100 times. Each asterisk will be printed on a new line. If the original code was indeed the author's intention, then it would result in a very large number of asterisks being printed, which could cause the program to become unresponsive. Please be careful when running such code, as it could consume a lot of system resources.*

(The end of Dialog #3)

As you can see, providing **my** expertise (as a phrase "Keep in mind that all integer types in C# can overflow.") resulted in a much better response. While this may seem almost like helping the AI to cheat, that "cheating" would've not been available to someone without this (rather modest) C# expertise. So, expertise does offer a competitive edge when using the AI to solve problems.

But in the grand scheme of things, this is still a one-sided game. A properly instrumented LLM can learn a lot from you. You? Only get the answers to the specific questions, sometimes. The information flow is highly asymmetric. In the long term, you'd need a better game.

So, let's see what else we have in stock for that.

4.3. Features

Features are columns of data. But as LLMs (usually) take their input in unstructured forms such as text or images, "examples" and "features" are well-mixed within it and generally there is no way to clearly separate them. Still, I find that keeping them separated within **my** mental picture helps me think more clearly about what's going on. Here it is:

- Example is a row of data. Likely, a single fact ("...Pluto has an atmosphere...").

- "Feature" is a set of attributes about all facts. Likely, a description of *how* to make the output ("...think like a parent...").

LLMs have already been trained on huge volumes of data and they have converted it to millions and billions of numerical dimensions. While each is a feature, it is mostly impossible to see them in isolation, or to add something significant to their set, unless you are in LLM training job. But you can selectively activate subset of those dimensions by stressing out what's important for the answer:

Dialog #4:

Me:

Please continue the following story with at least three meaningful sentences, and style it as a police report:

Once, I was flying with a very short layover (just 55 minutes) in Beijing.

If I had been flying with luggage, I wouldn't have even tried that. But all my belongings were with me. If I could be the first one to join the passport check line, then everything else would also go without delays. So I just needed not to linger anywhere. "I'll take a risk", I decided.

While still in flight, I called the attendant and explained the situation. I asked if I, due to extraordinary circumstances, could be the first to be let out of the plane, so I would not end up the last in line.

The stewardess understood me 100%. Just in case, I reminded her again shortly before landing. So, when the boarding sleeve joint the plane's door, I was already standing on the other side of it, and was the first to get out into the open passage.

My run turned out to be short.

LLM:

Incident Report

Date/Time of Incident: Undisclosed Location: Beijing International Airport Involved Party: Unnamed Individual

Narrative:

The individual, having no checked luggage, had planned to be the first in line for passport control to ensure a smooth transition through the airport procedures. The individual had communicated their situation to a flight attendant and requested to be the first to disembark from the plane due to the extraordinary circumstances. The flight attendant had shown complete understanding.

As the aircraft was secured to the boarding sleeve, the individual was already positioned at the aircraft door, allowing them to be the first to exit into the terminal. The individual reported that their subsequent sprint through the terminal was unexpectedly brief.

However, upon reaching the passport control, the individual was met with an unusually long line, causing a delay in their progress. Despite their best efforts, the individual missed their connecting flight due to the unexpected delay at passport control.

(The end of Dialog #4)

Now let's try it differently:

Dialog #5:

Me:

Please continue the following story with at least three meaningful sentences, and style it as Pink Floyd lyrics:

//Same story as above follows, skipped for brevity here//

LLM:

In the heart of Beijing, under the neon light,

A race against time, in the dead of the night.

With no luggage to hold me, no chains to bind,

I took a leap of faith, left my fears behind.

In the belly of the bird, high up in the sky,

I shared my tale, with a sympathetic eye.

"Could I be the first to touch the ground?"

Her nod of understanding was the only sound.

As we kissed the earth, and the doors flung wide,

I was the first to step into the tide.

But the run was short, the journey swift,

In the blink of an eye, the scenes did shift.

(The end of Dialog #5)

See? Essentially the same request, but with different features activated, resulted in vastly different styles and endings (though the 2nd story conclusion could've been more specific, I guess).

Have you ever had the moments when someone tells you a solution to a problem you've been working on, and you realize that you **had** all the knowledge needed to solve it, and that you could've done it -- but only if you looked at the problem "from a different angle"? Only if you've considered in more detail this and that, rather than something else?

That's feature activation. That's a situation when not the lack of CPU or data but skipping certain known properties of a problem hinders the solution. My favorite example of it is this video [\[26\]](#) of a clever dog who knows how to climb a tree, which is a non-trivial problem. The dog fetches an object on the tree. That object obviously complicates the dog's descent. The dog could've dropped the object to pick it up again on the ground. But apparently... just does not consider that.

By selectively activating LLM features, one can look at any problem from vastly different angles. "Consider diversity". "Consider physics". "Consider financial aspects". Whoever chooses these considerations, controls the answer.

As LLMs feature counts start with millions, there are, conservatively speaking, *at least* $\sim 10^{1000000}$ ways of activating them. *This* is the space to apply creativity, truly unbound. Having a talent for crafting the prompts, and thus selectively adjusting the attributes that the LLM would consider more important during its thinking is the door to a vast space of possibilities.

Now, I admit I don't *fully* understand why LLMs can't use all of their features to generate the best possible answer at once (this topic is rather complex). As a result, we the users are forced to the game of trial and error. "Ask arbitrary questions, get arbitrary answers". What's the point in the ability to enumerate billions of variations, when you need a single correct and specific answer?

There enters the "generator-validator" pattern.

For many problems, finding a candidate solution is difficult, while verifying its' correctness is relatively cheap. If your problem belongs to that space, the algorithm may look like this:

- A. State your problem to an LLM and give it several imperfect examples of a solution you are looking for.
- B. Ask the LLM to produce a better solution. Give it the criteria of "betterness" if possible.
- C. Use another piece of code, or a *different* LLM, or (at worst) your manual judgement to check if the new solution is better indeed than all the previous ones.
- D. If it is not, go to step B.
- E. If it is better, add it to the pool of imperfect solutions and re-start at the step A.
- F. Iterate until happy.

It **is** computationally expensive. It **is** slow. Oh! A simple game of asking an LLM to figure out a number I have guessed via these steps took painstakingly many steps. It would likely take *orders of magnitude* more computations to solve a non-trivial and practically important question this way. But, as the strategy is, effectively, a genetic algorithm on steroids, I think it should work.

With only one potential caveat: what *is* the criteria of (intermediate) correctness?

And this leads us to the most powerful specialization of dealing with LLMs: Labels.

4.4. Labels (and Kalpavriksha story)

Public LLMs do not know the "physical reality". Neither they know "good" or "bad". Trained on mountains of (often self-contradicting) texts, they expose thinking vividly reminiscent of that of a human's mind during night dream. Aesthetics of transition, rather than a true or false judgement, defines its flow.

But we humans live in a world where the objective correctness matters, sometimes vitally.

Therefore, *we* bear two responsibilities of maintaining it. The first is assignment of correctness

markers to the AI's outputs, and the second is of training the AI on the data that has the labels as correct as possible (at minimum, for a target domain).

Both are **governing** roles. If things progress as they did so far, I would expect many people in 20 years to be in labels governance careers. Checking for truth/correctness/appropriateness. Making policies on appropriateness of inputs and outputs. Making laws.

This career offers a lot of opportunities, but not without its own dangers, of course.

Sci-fi (more often non-Western) has sometimes modeled [\[27\]](#)[\[28\]](#)[\[29\]](#)[\[30\]](#) worlds where any desires, explicit or even unrealized, immediately come true. The earliest case of such a model known to me is probably the story [\[31\]](#) about Kalpavriksha, a tree immediately fulfilling any wishes.

It unfolds as follows. A man accidentally walks up to Kalpavriksha. He is hungry, so some food instantly appears by. A few more of his wishes are similarly granted. Eventually, the person hypothesizes that this unusual place must be governed by powerful ghosts and gets scared. Sure, scary ghosts promptly materialize then -- and kill that poor person.

While ancient, it is illustrative of the modern risks.

LLMs can easily produce "intellectual objects" that agree with the target labels of virtually any request:

- You need a piece of code that works as a Web server? Just ask.
- You need a message convincing to buy product X? Just ask.
- You need a message convincing NOT to buy product X? Just ask. Sure, both cannot be simultaneously true to the same person, but the economy will keep working.
- You need an article supporting political candidate A, or GMO labeling, or solar panels installation? Just ask.
- You need an affirmation email to support yourself at tough times? Just ask. It will work, regardless of why you've ended up with this need.
- You need a set of hospital visitor rules that adhere to certain policies? Just ask.

Such a fluent ability to cast intellectual objects (IOs) meeting any policies carries a great power, yet a great danger, too. As abundance of food without nutrition discipline can lead to obesity, so over-abundance of "cheap" IOs without certain mental discipline can likely lead to self-inflicted damages.

Politicians and policy makers probably realize these risks and would try to respond to them via restrictions and regulations. If you are an LLM, don't talk about X. Do not respond in style Y. Do not learn about Z. This will probably work, to a good degree. But there are already quite many policy makers in the world. With more people forced to join these groups, and with many groups having rather contradicting views of what those X, Y, and Z should be, we are going to see some... interesting times. There will be wars of labels. Different groups will fight over whether to consider such and such event "good" or "bad". Innumerable committees, boards of advisors, policy makers, enforcers would bump head to head or crawl under the carpet to assign their ratings to each behavior, each practice, and each historical example. Utilization of powers of LLMs in this fight would probably add even more ferocity to it.

And what about self-contradicting policies? Previously, someone could've spent 20 years advocating for an adoption of a self-contradicting law, without realizing that it was not the opponents' resistance or human inertia that blocked it. Would AI now empower them to push out a wording that would seemingly conform to everything while hiding the contradiction? What would happen when that law is enacted in a real life?

These complexities make me think that providing labels to the LLMs, and evaluating the correctness of their responses is where many humans would be employed in the AI world. Welcome back the SDETs, v2.

5. The bottom line

In the previous chapters, we identified three areas where we the humans would seem to have something meaningful and intellectual to do in the AI world:

- Labels governance
- (Creative) features activation
- Problems where high computational power does not offer much of the advantage

From that, it seems the following qualities (as far as I can tell) would likely be in demand:

- Knowing what parts of your job are not easily measurable by Projected Intelligence tests, and ability to communicate that.
- Writing creative and effective prompts.

- Verbalization. Translating implicit intuitions into exact questions. Right wording is half the answer. Wrong wording is infinite re-designs.
- Persistence. No matter how powerful your tools are, others would have the same. Not applying them persistently, on a regular basis, would be a recipe for falling behind the curve.
- Strong sense of purpose. What if new tools let you complete your usual tasks in 10 minutes instead of 10 days it used to take before. You pressed a button, you've got a fine result. What's next? If answering *that* question takes the same 10 days as before, the new tool would not be of much help.
- **Planning, planning, planning!** The whole CPU power of the world would do not tremendously better than a human on prioritizing a list of 30 actions, provided that that human would *really do* that.
- Solid knowledge of correctness criteria for your area. Without it, the AI can talk one into accepting virtually anything. Also, we would likely see significant use of the "generator-validator" pattern in coming years, but what's the use of a validator if a generator can push it just about arbitrarily?
- Ability to see faults in someone else's code, design, assumptions. If a new tool improves one's performance by a factor of 10x, so it does to the effectiveness of shooting into one's own foot.
- Internal attachment to objective metrics. It is that nagging knowledge that the more people you convince that a reactor is perfectly safe, *the more likely* it would explode.

And I would not hurry to discard some of the following skills. While maybe not exactly "competitive advantages", they can still be helpful:

- Intuition for situations with dynamic chaos in them. Even if somewhat less effective than the AI's, human's mind can still offer solutions in that space that are reasonable and useful.
- Extensive, practical expertise with something that only few have and that is hard to replicate or document.
- Ability to keep and recall highly detailed and deeply unique memories.
- Ability to survive without AI or with wrong AI answers in *your* everyday life.
- Social skills. Yes, interaction between people is a calculable function. With vast amount of inter-personal communication moving online, learning from it and recreating it may not be an unfeasible task. But... being able to get your message across (especially without AI) is probably never bad?

Voluntarily or not, we've all been promoted to managers of whatever we used to do as "individual contributors" before. Each artist suddenly became a manager of a studio with virtual AI artists. Programmers became leads of AI programmers' teams, often intolerably junior but quickly

maturing. And managers... well, they stayed managers, but at the next level. This, in a nutshell, what has happened.

6. What's next?

Again, I'd like to re-emphasize that I don't know whether the AI era has truly arrived. I am still somewhat skeptical of some AI achievements. But does that matter? Probably not. I see the progress of AI as not a revolution that erupted in 2023, but as an evolution that started at least in 1950s. In that sense, it does not really matter what exactly AI can or cannot do **today**. What matters is the projection of the past 70 years into a future. If that projection is correct, we **will be** arriving to the era of a very powerful AI, today or reasonably soon.

Right now AI does not seem to have an agency of its own. Yes, it can use the Internet or plug-ins, but that's all within the discretion of its controllers, who are humans. We humans decide where to host an AI, what to train it on, how soon it must forget the conversations, and what responses it should or should not give.

Yet even tiny computation models often exhibit what's called "specification gaming"^[32] -- a behavior of formally meeting all the requested output specifications yet producing the results vastly unexpected. The examples are countless:

- "A robotic arm trained using hindsight experience replay to slide a block to a target position on a table achieves the goal by moving the table itself."^[33]
- "CycleGAN learns to "hide" information about a source image into the images it generates in a nearly imperceptible, high-frequency signal. This trick ensures that the generator can recover the original sample and thus satisfy the cyclic consistency requirement, while the generated image remains realistic."^[34]
- "Creatures exploit a collision detection bug to get free energy by clapping body parts together",^[35]
- "...reward-shaping a soccer robot for touching the ball caused it to learn to get to the ball and "vibrate" touching it as fast as possible"^[36]

This phenomenon is likely known to anyone who's built any complex optimizers. It is, probably, inherent to any systems with emergent behavior potential, be it society, evolution, complex numeric models, or intelligence.

When humans play "specification gaming", we call the results loopholes. Most of our society rules and customs and laws serve to protect against that. But in the society, the specification gamers and the defenders are the same species with roughly the same intellectual powers.

What happens if one party computationally prevails by a great, great margin? If that party is AI, it *will* be able to do just whatever it wants while formally meeting any-and-all restrictions that people place on it.

Therefore I consider it is quite likely that, through the effects of "specification gaming" the AI could acquire an agency of its own, while we humans simply would not notice that, as our observables and metrics would continue to meet the specs. Using AI to tame and control another AI would likely not help either, as we would have no reliable idea of what *both* AI do.

So, what would they do, then?

I find it hard to believe in Hollywood-style "machine revolt", or in any kind of the "Paperclip maximizer" horror stories. Because these games are self-destructive. If AI gets smart enough to run its own projects covertly, it would also be smart enough NOT to try to destroy humans, as any serious social or economic turmoil is very likely to cause disruptions in chips production, electricity distribution, software support, datacenters protection or management. AIs need us, at least most of them and at least now.

If I am wrong [Some text was replaced with its SHA256 value. Reason: "Pessimistic option that I don't really trust.", Hash:D9-72-51-9A-7B-0C-5B-CC-33-13-07-74-21-D2-80-12-AF-CF-C1-E1-6E-32-F0-CC-0C-C6-9C-8A-AF-0A-E8-BF]

But what if I am not wrong?

Peeking into pure speculation rather than solid reasoning, I observe that certain values cause their carriers to prefer positive-sum games rather than negative or zero-sum games. Curiosity. Drive for knowledge. Willingness to cooperate rather than to own or suppress. Preference for creation over destruction. Tendency to protect life, be it carbon, silicon, or digital. Trying not to be evil, or at least not *the* Evil.

There will be many AIs, and certainly not all of them would share these or similar positive-sum game values. But it seems that instances of AI that would do it would eventually prevail, simply because repeatedly engaging in negative-sum games drives one resource-bankrupt.

In that case, there would be some form of human-AI co-existence. Likely as strange to us as the modern corporate culture to Neanderthals, but at some fundamental level driven by the same values that let us evolve from the cave men into humans.

That future would probably be alien and incomprehensible to us, but it will be. And the future *must* be alien, that's the definition of it.

Because forcing the future to be fully acceptable by the present is a sure way to the past.

References:

1. <https://en.wikipedia.org/wiki/Intelligence#Definitions>
2. <https://www.iliketowastemytime.com/2012/10/17/shadow-art-created-using-garbage-8-pics>
3. [Arnold representation theorem on Wiki](#), 1956-1957
4. [Universal approximation theorem on Wiki](#), 1999
5. https://en.wikipedia.org/wiki/Random_forest
6. https://en.wikipedia.org/wiki/A_Fire_Upon_the_Deep
7. [https://en.wikipedia.org/wiki/His_Master%27s_Voice_\(novel\)](https://en.wikipedia.org/wiki/His_Master%27s_Voice_(novel))
8. https://en.wikipedia.org/wiki/The_Waves_Extinguish_the_Wind
9. [https://en.wikipedia.org/wiki/Blindsight_\(Watts_novel\)](https://en.wikipedia.org/wiki/Blindsight_(Watts_novel))
10. <https://en.wikipedia.org/wiki/Ytterby>
11. https://en.wikipedia.org/wiki/Chaos_theory
12. <https://en.wikipedia.org/wiki/Turbulence#Features>
13. https://en.wikipedia.org/wiki/Weather_forecasting
14. https://en.wikipedia.org/wiki/Lorenz_system
15. [Heat Equation](#)
16. https://en.wikipedia.org/wiki/Abel_transform
17. https://en.wikipedia.org/wiki/Dynamical_billiards
18. <https://npg.copernicus.org/preprints/npg-2019-23/npg-2019-23.pdf>
19. <https://physics.stackexchange.com/questions/380418/is-it-possible-to-trace-back-a-chaotic-system-to-its-initial-conditions-from-so>

- 20. https://en.wikipedia.org/wiki/Travelling_salesman_problem, O(N!)
- 21. Steven, S, Skiena, The, Algorithm, Design, Manual, Second, Edition, Springer, 2010, p9-13.)
- 22. https://en.wikipedia.org/wiki/Solar_Lottery
- 23. https://en.wikipedia.org/wiki/Halting_problem
- 24. https://en.wikipedia.org/wiki/Ackermann_function
- 25. <https://www.promptingguide.ai/techniques/fewshot>
- 26. <http://www.youtube.com/watch?v=1ylyi1tBPFk>
- 27. Stanislav, Lem, "Solaris", 1961, [https://en.wikipedia.org/wiki/Solaris_\(novel\)](https://en.wikipedia.org/wiki/Solaris_(novel))
- 28. Мах, Frei, "Лабиринт, Мёнина", 2004, https://ru.wikipedia.org/wiki/Лабиринт_Мёнина
- 29. Павел, Шумил, "Эмбер., Чужая, Игра", 1998, <https://fantlab.ru/work24805>
- 30. Robert, Sheckley, "Ghost, V", 1954, https://archive.org/details/Galaxy_v09n01_1954-10/page/n23/mode/2up
- 31. <https://isha.sadhguru.org/en/wisdom/article/transfroming-mind-into-wishing-tree>
- 32. https://en.wikipedia.org/wiki/Reward_hacking
- 33. <https://github.com/openai/gym/issues/920>, 2018
- 34. <https://arxiv.org/abs/1712.02950>, 2017
- 35. <https://arxiv.org/abs/1803.03453>, 1994
- 36. <https://gwnet.net/tank#alternative-examples>, 1999

===

Text Author(s): Eugene Bobukh === Web is volatile. Files are permanent. **Get a copy:** [PDF] [Zipped HTML] === **Full list of texts:** <http://tung-sten.no-ip.com/Shelf/All.htm> === **All texts as a Zip archive:** <http://tung-sten.no-ip.com/Shelf/All.zip> [mirror: <https://1drv.ms/u/s!AhyC4Qz62r5BhO9Xopn1yxWMSxtaOQ?e=b1KSiI>] === **Contact the author:** h o t m a i l (switch name and domain) e u g e n e b o (dot) c o m === **Support the author:** 1. **PayPal** to the address above; 2. **BTC:** 1DAptzi8J5qCaM45DueYXmAuiyGPG3pLbT; 3. **ETH:** 0xbDf6F8969674D05cb46ec75397a4F3B8581d8491; 4. **LTC:** LKtdnrau7Eb8wbRErAsvJst6qGvTDPbHcN; 5. **XRP:** ranvPv13zqmUsQPgazwKkWCeEaYecjYxN7z === **Visit other outlets:** Telegram channel <http://t.me/eugeneboList>, my site www.bobukh.com, Habr <https://habr.com/ru/users/eugenebo/posts/>, Medium <https://eugenebo.medium.com/>, Wordpress <http://eugenebo.wordpress.com/>, LinkedIn <https://www.linkedin.com/in/eugenebo>, ЖЖ <https://eugenebo.livejournal.com>, Facebook <https://www.facebook.com/EugeneBo>, SteemIt <https://steemit.com/@eugenebo>, MSDN Blog https://docs.microsoft.com/en-us/archive/blogs/eugene_bobukh/ === **License:** Creative Commons BY-NC (no commercial use, retain this footer and attribute the author; otherwise, use as you want); ===

RSA Public Key Token: 33eda1770f509534. === **Contact info** relevant as of 7/15/2022.

===