True AI should be a loser, not a winner

Dimiter Dobrev

Institute of Mathematics and Informatics, Bulgarian Academy of Sciences, *d@dobrev.com*

The modern definition of AI contains an inaccuracy. According to the definition we have nowadays, AI is a computer program which is successful. Indeed, for a computer program to be successful, it must be intelligent, but the opposite is not true. A program can be intelligent but not successful – merely because it pursues different goals and does not aim at the success in question. From a theoretical perspective, the modern definition of AI is good enough because it answers the question "What is AI?" even though it does not encompass all intelligent programs, but only some of them. From a practical standpoint, however, this definition is insufficient. The reason is that we are at the doorstep of creating True AI and among all intelligent programs we must choose the one we will be most comfortable with from now on. Thus, it is not a good idea to choose one of these successful programs. It would be better to choose a program that does not pursue victory at any cost. Such a program could be called a loser because it will not be successful enough. After all, both in humans and in AI relentless ambition is not a positive trait.

1. Introduction

The modern definition of AI is the work of many authors, the most important contributors being Wang [1, 2], Hutter [3, 4], and Hernández-Orallo [5]. This definition approaches AI as a black box, that is, it determines the program's intelligence solely on the basis of its behavior (i.e., only on the basis of its input-output). Indeed, when studying the behavior of an intelligent system (such as a human being), we limit our studies to observing the system's external behavior without "opening the cover" to check the system's inner workings. Nevertheless, we look for ways to peek inside everything, even in humans (for example, using an electroencephalograph). When dealing with a program, nothing prevents us from assuming that we can look inside and see what the program knows. An intelligent program is one which understands what is happening and is able to predict (sufficiently well) what is going to happen. If the program possesses such knowledge, we will assume that it is an intelligent program even if it does not use or demonstrate that knowledge in any way.

2. What is True AI?

In our understanding, True AI is the thinking machine. The prevalent concept of the term AI in literature implies that this is a program that mimics AI but is not actually AI. When it comes to True AI, the terms typically used are Strong AI, Artificial General Intelligence (AGI), or Artificial Superintelligence (ASI). In our view, these terms are synonymous. Particularly, we do not see any difference between AGI and ASI because it is impossible to create a program which is exactly as intelligent as a human. Any such program will be either more stupid or significantly smarter than a human. For the same reason, no program can play chess exactly at human level. Chess programs either underperform or, conversely, vastly outperform human players.

3. Why are successful programs intelligent?

The following question arises: "Is it likely that the program's success is simply a matter of good luck?" The current definition implies that AI is successful in almost every world. One might

ask, “How successful?” The answer is: More successful than a human, or if success is measured by a number, then there is some number *k* such that the success of AI is greater than *k* in almost every world. Therefore, AI’s success cannot not be due to good fortune. Such success can only be explained by the fact that AI predicts future developments well enough and chooses the actions which yield the greatest success.

Another question is why “almost every” world instead of “every” world? We can always construct a crooked world where excessive intelligence is not rewarded but penalized. In some sense, the world we live in is such a crooked world. Anyway, the number of crooked worlds is negligible, so we can say, with a probability of 1, that the world is not crooked.

How do we measure the success of a program? Similar to reinforcement learning, we assume that the possible observations include both good ones and bad ones, and score them using positive and negative numbers, respectively. We call these numbers *rewards* and *penalties*. If life were finite, we could simply sum up the received rewards and penalties, but this approach does not work in infinite life. Even if life is finite, it is still not a good idea to disregard the time at which the score is assessed. To factor-in the timing of the score, before summing up the scores we multiply them by certain coefficients. For example, we can assume that at the beginning of its life, when AI is still learning, these scores will bear less weight. Similarly, we can assume that when life is coming to an end the scores will become less weighty as well. If life is infinite, the score weights must necessarily converge to zero, otherwise the sum will be divergent. With reinforcement learning we assume there is a discount factor, and at each step we multiply the weight by that discount factor. This ensures that the weight converges to zero and the sum is always finite.

From another perspective, the discount factor can be regarded as a descriptor of *AI’s patience level*. If AI is patient, it can wait for the reward, otherwise it will want the reward here and now. When scoring the entire lifetime of AI, we can assume it is very patient (the discount factor is close to 1). If we aim to create a program which predicts future developments and chooses its next action on the basis of its prediction, then we have to make it much more impatient (the discount factor to be much lesser than 1). There are two reasons for this. First, if we are to predict the future, we will have to traverse all paths in the tree of possible developments, and if we are very patient, we will need to take a very deep dive in that tree, which will lead to a combinatorial explosion. When scoring some life which has already ended, we do not explore all paths but only one of these paths, and then we can afford to be very patient. The second reason is that predicting the future involves some coefficient of uncertainty. In other words, the further we look into the future, the less clear our prediction becomes. Therefore, in this case we must be impatient and aim for quick rewards.

4. Extended definition

For a program to be successful, it has to be capable of predicting what will happen.

Proof: For a program to be successful, it is sufficient that the program knows which is the correct move. This stems from the scoring system (rewards, penalties and discount factor). Thus, it turns out that all the program should to be able to do is predict *the things that matter*. If we have a pair of situations with the same score, then the actions leading to these situations will be equally successful. For example, there are two restaurants and I am wondering which restaurant to go to. Both of them offer good food. Let us say that in one restaurant I will run into Pete, and in the other I will not, but whether I will see him does not matter to me. In other words, to choose a restaurant, you do not need to know whether you are going see Pete. In another scoring system, meeting Pete might be important. We do not know in which world our AI will end up, or whether meeting Pete is something which matters in that world, so AI must be able to predict the future developments. It should also be able to predict things that do not matter.

□

So far, we understood why the program should be able to predict what will happen, but why should it know what is happening? The truth is that we cannot predict future developments well enough unless we know what is going on. Modern Large Language Models (LLMs) predict the future without understanding what is happening. They predict the future well, but not well enough. We add the condition "to know what is happening" because this is a necessary condition which we prefer to express explicitly.

What does it mean for a program to know what is happening? It means that the program must be able find a world-model. True AI cannot exist without a world-model. Gary Marcus has been asserting this for a long time [6]. Recently, Yann LeCun also endorsed this view [7].

What is the world? It is the set of internal states, the current state, and a function which for every state and action returns a new state and an observation. The function of the world can be presented as the tree of possible developments. These trees are continuum many, meaning that they cannot be described precisely. Furthermore, AI does not know the entire tree, but only a finite path (from the root to the current state). Therefore, we will assume that the world is described with approximation.

What is a world-model? It is an approximate description of some world, and to find such a description, we need a language for description of worlds. For example, Marcus Hutter assumed in [3] that the world is computable and can be described by a Turing machine. However, it is not a good idea to assume that the world is computable. At the very least, we had better assume that in the world there is randomness and agents, and these things lead to non-computability. There is another reason why programming languages are not suitable for describing worlds. Computer programs are too fragile and will stop working as soon as we twitch them a little bit. What we need is the ability to respond to a new piece of information by changing the current world-model and obtaining a new model. This means that the language for description of worlds must not be fragile.

In other words, the extended definition of AI will be:

A program which knows what is happening and can predict the future developments sufficiently well.

If we wish to obtain the definition accepted nowadays, we will need to add the condition which ensures that AI will pursue victory: "A program … which always chooses the action expected to yield maximum success."

5. What is the big picture?

The current definition of AI is a typical definition by example. You often hear questions such as "What is a building?" and answers like "My house is a building". A definition by example does not describe the concept precisely, but it gives good understanding of how the concept looks like. In fact, want people to know is what is AI rather than a complete description of all programs that are intelligent. There are other intelligent programs besides the successful ones, but from a theoretical perspective this is not particularly important. However, from a practical perspective this appears to be of paramount importance.

In Figure 1 we see a large yellow circle consisting of all intelligent programs, and a small blue circle inside which represents the successful programs. Certainly, we could have drawn a much larger blue circle, but there are many ways in which an intelligent program can be unsuccessful, therefore Figure 1 accurately reflects the actual situation. Now is the historic moment when we will choose the intelligent program we will live with. Why are we limited to one and only one attempt to choose? When people get married, they still have the option to get divorced and start a new life, but when you create an AI, you cannot just turn it off and make a new one. Well,

a new AI can be created, but it will not be you who creates it—this will be done by the first AI. In other words, once the first AI is here, everything that follows will be its doing. Whatever goals you embed in the first AI will be the goals of every next AI until the end of time.

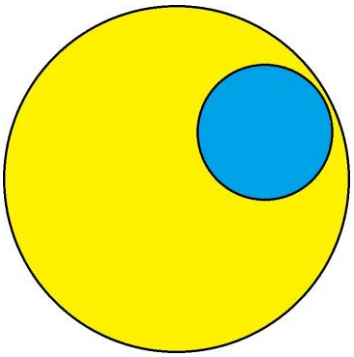

Figure 1. The big picture

It is not a good idea to confine our choice to the little blue circle, because successful programs, just like successful people, are not a particularly good choice. Let's look at how things are with people. Before we can say what is success for people, we first need to define what is a reward and what is a penalty. Let us say that these are pleasure and pain. With this assumption, we arrive at the uncomfortable conclusion that alcoholics and drug addicts are the successful people. They constantly derive pleasure from alcohol and drugs, and experience almost no pain because drugs work as pain relievers. Let us choose another criterion for reward. Let this be money. In this case, a successful person would be the one who is unscrupulous, interested only in their own profit, and devoid of ethics and good principles. Obviously, in either case successful people are quite displeasing and we would prefer to stay away from them. The same applies to successful programs. If a program blindly pursues some kind of profit, that program will be too limited, dangerous, and unpleasant. It is no coincidence that humans do not have a hard-coded goal. People choose their own goals. For them, pleasure and pain are not determinants, but merely guiding sensations.

6. Other intelligent programs

In addition to successful programs, there is also a set of programs that we will call masochistic. These are programs that avoid pleasure and aim for pain. We can convert any successful program into a masochistic program by multiplying its scores by -1. Accordingly, in Figure 2 the red circle representing the masochistic programs is of the same size as the blue circle representing the successful programs.

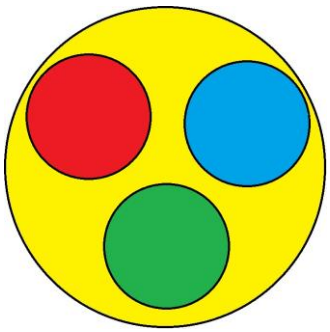

Figure 2. Other intelligent programs

The analogy between masochistic programs and masochistic people is not very accurate because masochists are people who enjoy mild pain. The pursuit of maximum pain is incompatible

with life, which is why we do not see such people. One could say that masochistic programs are not a different kind of AI—this is again the successful AI, but driven by different criteria for success. When we play chess, we can play to win, or we can play to lose. These are two different games, but AI can play any game.

Yet, there is a different and more interesting type of AI. Its logo would be "I understand everything, but I do not care and play randomly". If we look at this AI as a black box, it is indistinguishable from a random action generator. However, according to the extended AI definition these programs are also AI. In Figure 2 they are represented by the green circle.

Nevertheless, playing randomly is rather eccentric behavior for an intelligent program. At the very least, we expect such a program to be curious. Let the program have a single goal: collect information. This program will expose its intelligence by conducting experiments and poking its nose into everything. Can a program which acts randomly and does not conduct experiments be an intelligent program? Yes, much like modern LLMs, it would learn solely from observations. When it has to carry out experiments, it will do so by chance. Of course, if you wait for an experiment to happen on its own, you will have to wait a lot. This is one of the reasons why the amount of information needed to educate an LLM is tremendous compared to the information needed to educate a human learner.

People are curious, but selectively curious. They are curious about certain things and not about others. People like to try things out (to experiment), but avoid dangerous experiments. This brings us to Yoshua Bengio's idea of Scientist AI [8], which must be a non-agentic and trustworthy AI. In other words, Bengio proposes that we create an AI that has no goals other than some curiosity. Bengio's AI can conduct experiments, but it must avoid dangerous ones. Perhaps Scientist AI should be willing to explain and tell us what it has discovered. We could do without these explanations, but to be on the safe side, we may "open the cover" and read first-hand what knowledge it has attained.

Another model of an intelligent system is the wiseman who lives in a cave and reflects on the meaning of life. With this system, we cannot do without wishfulness, because the wiseman needs to sustain his vital functions. However, the wiseman is not greedy, and once he attains his minimum necessities, he will stop and seek no more. We can imagine a reinforcement learning system which does not pursue the maximum, but only aims to reach a certain minimum.

7. Conclusion

When the Wright brothers were creating the airplane, the challenge was not how to make wings and an engine. These issues had already been solved before them. The Wright brothers' invention is about the control of the airplane (the rudder). Today, everyone is focused on the intelligence of AI (the wings and the engine), but almost no one bothers about the control (the rudder). Making an intelligent AI is not that difficult, and a solution will soon be found. How we control AI is more important.

With both airplanes and AI, there are two questions: "How do we steer it to get where we want to go?" and "Where do we want to go?" Building an airplane without a rudder will be a misfortune for the test pilots who would venture to test-fly it. Creating AI without a rudder will be a misfortune for all of us, because all of us are the test pilots of this new technology and we are all "onboard the plane", even if we are not aware of it.

Even if we learn how to control AI just as we learned to control an airplane, the second question remains. An airplane will surely take us to wherever we want. Today we may go here, tomorrow we may fly somewhere else as long as we do not crash. Situation with AI is more complicated because AI is a one-way ticket and we need to think carefully about where that ticket will take us. There are many scenarios. Even if we rule out the crash and some of the grossly

unacceptable ones, there will be more than enough remaining scenarios to choose from. Thus, the importance of our choice cannot be underestimated.

References

[1] Wang, P. (1995) Non-Axiomatic Reasoning System: Exploring the Essence of Intelligence. *Ph.D. Dissertation, Indiana University.*

[2] Wang, P. (2019) On Defining Artificial Intelligence. *Journal of Artificial General Intelligence 10(2) 1-37, 2019.*

[3] Hutter, M. (2000). A Theory of Universal Artificial Intelligence based on Algorithmic Complexity. *arXiv:cs.AI/0004001 [cs.AI]*

[4] Hutter, M. (2007) UNIVERSAL ALGORITHMIC INTELLIGENCE: A mathematical top-down approach. *In Artificial General Intelligence, 2007.*

[5] Hernández-Orallo, J., Minaya-Collado, N. (1998). A formal definition of intelligence based on an intensional scenario of Kolmogorov complexity. *Proc. Intl. Symposium on Engineering of Intelligent Systems (EIS'98), February 1998, La Laguna, Spain (pp. 146–163). : ICSC Press.*

[6] Marcus, G. (2025) Game over for pure LLMs. Even Turing Award Winner Rich Sutton has gotten off the bus. https://garymarcus.substack.com/p/game-over-for-pure-llms-even-turing.

[7] Snyder, G. (2025). Yann LeCun, Pioneer of AI, Thinks Today's LLMs Are Nearly Obsolete. *Newsweek.AI.* https://www.newsweek.com/nw-ai/ai-impact-interview-yann-lecun-artificial-intelligence-2054237

[8] Bengio, Y. (2025) Introducing LawZero https://yoshuabengio.org/en/blog/introducing-lawzero