

Evaluating Understanding

Jordi Bieger¹ & Kristinn R. Thórisson^{1,2}

¹ Center for Analysis and Design of Intelligent Agents,
School of Computer Science, Reykjavik University, Iceland

² Icelandic Institute for Intelligent Machines, Iceland
{jordi13, thorisson}@ru.is

Abstract

Understanding is an important aspect of human intelligence that has taken the back seat in many approaches to AI. While results in automation can be achieved without it, we argue that understanding is especially important for general-purpose systems. It is doubtful that we could classify systems as having general intelligence if they don't really understand their tasks, environment, and world, and thus it is important for us to verify the level of understanding of any system intended to strive for generality and autonomy. Understanding goes beyond “good performance” on a range of dimensions—and might even be said to be necessary (but not necessarily sufficient) for general intelligence. In any case, understanding is likely to become a major subject of inquiry in AI in the coming decades. But how would we test for understanding?

Because understanding is an internal property of a system, evaluation can be difficult. We propose four kinds of tests for understanding: prediction, goal achievement, explanation, and (re)creation.

1 Introduction

Understanding is a core aspect of intelligence. It implies an ability to not only know facts and perform skills by rote memorization (“blindly”), but to be able to reason about them, dissect them, achieve goals with them, and generally *explain* them. Despite the importance of the concept of understanding, relatively little attention has been paid to it in the field of AI, and even in philosophy [Sloman, 1985].

In prior work we have characterized understanding as the construction and use of a causal-relational model that incorporates causal, as well as other relevant, relations of the phenomenon to be understood [Thórisson *et al.*, 2016b]. For some tasks, understanding is not strictly necessary; memorization can for instance work fine in tasks that don't change, and knowledge of mere correlations (as opposed to causation) is often sufficient for mere prediction. But for general-purpose AI or artificial general intelligence (AGI) it is difficult to see how we could get away with a system having no understanding of its task, environment, or purpose.

A major hallmark of AGI is the ability to operate successfully and robustly in a wide range of task-environments¹, even ones that were not necessarily envisioned by the system's creators. Doing this requires more than memorization and learning of statistical patterns. Without an understanding of what made the system's knowledge true in the old situations, there is no way to systematically transfer and adapt that knowledge to new situations, and the system would break down, behave unpredictably, or need to start learning from scratch. AI systems that seemed to perform very well before can fail in ways that seem utterly baffling to an observer, because they are oversensitive to changes, no matter how large or tiny, that were not present during learning/training [Szegedy *et al.*, 2013 12 20; Nguyen *et al.*, 2015].

Understanding is not only important for an AI/AGI system itself, to be more capable in its endeavors, but also for our own trust in such systems: There should be some way for us to ascertain that they understand what we entrust them with, that they understand their own limitations, and that they can acquire better understanding when needed. Almost by definition, we cannot test every possible situation such a system could find itself in — and certainly not future circumstances that are unknown at the system's design time. We may, however, want to reliably know how it might adapt to various *kinds* of future circumstances, events, and conditions², and this is where understanding enters the picture. This paper addresses the question of how we can evaluate whether an AI system does indeed understand.

Note that this is different from, but related to, the question of how an observer or system designer can understand the behavior and decisions of an AI system. The difference lies in who is attempting to do the understanding—we or the AI system—and what is being understood — the AI's decisions or some phenomenon of interest to the AI. A similarity is that we are trying to learn something about internal mental

¹We consider the separation of task from the environment in which it is performed to be a somewhat arbitrary one, so we use the term task-environment to refer to all aspects of a job that a machine would be doing [Thórisson *et al.*, 2016a].

²Whether or not future doomsday scenarios involving superintelligent machines are a remote possibility or a guaranteed eventuality [Bostrom, 2014], it is important to be able to ascertain that a highly capable system truly understands its (and our) goals [Steunebrink *et al.*, 2016].

processes of the AI system in order to increase our trust in it. Because of the need to trust AI systems that are increasingly affecting the way we perform important, high-impact, and safety-critical tasks, the fields interest in “explainable AI” (XAI) has grown stronger recently [Gunning, 2016 08 10].

In our approach to evaluating understanding, explanation is one of four criteria we should test for: In addition to making good predictions of a phenomenon, a system that truly understands it should also be able to achieve goals with respect to it, should be able to explain it, and eventually should be able to (re)create it.³

In the next section we will describe the various background concepts required for talking about the evaluation of understanding in an AI system. Section 3 will then discuss four aspects of understanding and their evaluation: prediction (Section 3.1), goal achievement (Section 3.2), explanation (Section 3.3) and (re)creation (Section 3.4). We will then briefly reflect on the difference between evaluating understanding of a phenomenon and evaluating the ability to acquire understanding in general in Section 4. Finally, we will offer some concluding remarks and future research directions in Section 5.

2 Background Concepts

2.1 System-World Interaction

For the purposes of this paper, we take a somewhat “dualistic” view of the interaction between an intelligent system and its world⁴ [Thórisson *et al.*, 2016a]. Intelligent systems continually receive inputs/observations from their environment and send outputs/actions back. Some of the system’s inputs may receive special treatment — e.g. as feedback or a reward signal.

The world W consists of a set of variables V , dynamics functions F , an initial state S_0 , domains D of possible values and ranges for those variables, and a possibly empty set of invariant relations between the variables R : $W = \langle V, F, S_0, D, R \rangle$. The variables $V = \{v_1, v_2, \dots, v_{\|V\|}\}$ represent all the things that may change or hold a particular value in the world. (A system’s dynamics can intuitively be thought of as its “laws of nature”.) As a whole, the dynamics may be viewed as an automatically executed function that periodically or continually transforms the world’s current state into the next: $S_{t+\delta} = F(S_t)$. The state-values of any element or variable in the task-environment at any point in time is determined by a causal chain, and we can visualize this in a

³By recreate we don’t mean *physical* recreation (which would put any part of the solar system out of reach of understanding) but rather to put forth a *necessary and sufficient model* of the whole phenomenon such that no aspect of it remains unexplained.

⁴The formulation of an intelligent system (or agent) interacting with the world (or environment) is most commonly used in control theory and reinforcement learning. However, it is a fully general formulation, that also covers traditional cases of e.g. supervised and unsupervised learning. Here the environment simply presents a (training) datum at each time step, the agent responds with a classification or prediction, and—in the case of supervised learning—the environment replies with the target outcome or an error signal.

directed acyclic graph (see Figure 2 the graph corresponding to the environment introduced in Section 2.3).

Each variable v may take on any value from the associated domain $d_v \in D$. For physical domains we can take the domain of each variable to be a subset of the real numbers. Invariant relations \mathfrak{R}_I are Boolean functions over variables that hold true in any state that the system will ever find itself in. In a closed system (with no outside influences) the domains and invariant relations are implicitly fully determined by F and S_0 . In an open system—where change may be caused externally—explicit definition of domains and relations can be used to restrict the range of possible interactions.

Environments are views or perspectives on the world. In their simplest form they can be characterized as slices or sub-spaces of the world, where all variables can take on a subset of the world’s variables, each variable’s domain is a subset of that variable’s domain in the world, and only the relevant dynamics and invariants are inherited.

2.2 Models, Phenomena, & Understanding

Our theory of understanding rests on the notion of minds, or agents, making *models* of phenomena *external* and *internal* to themselves. The models consist of many parts (except *atomic* models — the smallest possible), which are in turn (sub-)models that can be used and re-used to model other phenomena [Thórisson *et al.*, 2016b]. A model M of a phenomenon Φ is denoted M_Φ .

A phenomenon Φ similarly consists of a set of elements $\{\phi_1 \dots \phi_{\|\Phi\|} \in \Phi\}$, which can consist of other phenomena, variables, and relations \mathfrak{R}_Φ (causal, mereological, etc.). A phenomenon is any grouping of variables and relations in the world that we choose to group as such; $\Phi \subset W$. \mathfrak{R}_Φ couples elements of Φ with each other, and with those of other phenomena in the world [Thórisson *et al.*, 2016b]. These can be partitioned into inward facing relations \mathfrak{R}_Φ^{in} between element pairs $\phi_i, \phi_j \in \Phi$ and outward facing relations \mathfrak{R}_Φ^{out} between element pairs $\phi_i \in \Phi$ and $\psi_j \in W$. An agent’s understanding of a phenomenon Φ is perfect if the model(s) it possesses of Φ *accurately* and *completely* capture Φ ’s relations \mathfrak{R}_Φ .⁵ An agent whose models are only accurate for \mathfrak{R}_Φ^{in} understands Φ but not Φ ’s relation to other phenomena; an agent whose models are only accurate for \mathfrak{R}_Φ^{out} understands Φ ’s relation to other phenomena but will have limited or no understanding of Φ ’s internals.

M_Φ is thus a set containing models of a phenomenon Φ $\{m_1 \dots m_{\|M_\Phi\|} \in M_\Phi\}$. The closer the information structures $m_i \in M_\Phi$ represent elements (sub-parts) $\phi \in \Phi$, at any level of detail, including their couplings \mathfrak{R}_Φ , the better a system with models M_Φ understands Φ .

There is a correspondence between our theory of task-environments [Thórisson *et al.*, 2016a] and our theory of understanding [Thórisson *et al.*, 2016b]. The elements of a phenomenon are directly related to the elements of environments:

⁵For any complex phenomenon in a complex world, completeness of \mathfrak{R}_Φ^{out} is generally not to be expected, as this may be an extraordinarily large number. However, for any two phenomena Φ and Ψ that are related, if $\|\mathfrak{R}_\Phi^{out} \cap \mathfrak{R}_\Psi^{out}\| = \textit{small}$ then understanding \mathfrak{R}_Φ^{out} may not require a broad understanding of Ψ , even if $\|\mathfrak{R}_\Psi^{in}\| = \textit{large}$.

both have variables, can be hierarchically organized (into sub-environments or sub-phenomena), and contain some form of relations between these elements (\mathfrak{R}_Φ for phenomena and dynamics and invariants for environments). As such, we can use much of our ability to reason about task-environments to reason about phenomena for understanding, and vice versa.

2.3 Circuit Example

Throughout the paper we will use the example of understanding about electronic circuits (see Figure 1). An understanding of e.g. AND, OR and NOT gates can occur at the electronics level (top row of Figure 1) and include inward facing relations between subcomponents, or it can occur at the logic level where they have certain abstracted behaviors that can be linked together into larger systems, like an OR gate built out of AND and NOT gates or a full binary adder (bottom row of Figure 1). Below the electronics level lies the chemico-physical level, and above the logic level lies e.g. the context for any such circuit, its use in a larger system, for various purposes, etc.

The full binary adder serves as an example of a task-environment to be modeled using variables and causal relations between them (see Figure 2). The adder takes three binary inputs A, B and C, and produces two binary outputs that represent the sum of A, B and C, which can be 0 (00), 1 (01), 2 (10) or 3 (11). To make things more interesting and relate the example more closely to real world tasks, we have added delays to the circuit (this could be achieved using capacitors on the electronic level, or digital circuits—the exact method is immaterial for our purposes). Delays introduce a need for reasoning about time, for instance acting fast enough to achieve goals, predicting temporal behavior of the circuit, or explaining why one must wait for certain events to happen when starting from a particular state.

3 Signs of Understanding

Evaluating understanding, in our approach, is similar to evaluation of a human’s or animal’s ability to do anything: the more we test, and the more thoroughly we test, the more reliable and trustworthy our estimation of the tested ability will be.

To evaluate understanding of a phenomenon we will look at four different performance criteria of a purported understander:

- the ability to make predictions about the phenomenon
- the ability to achieve goals with respect to the phenomenon
- the ability to explain the phenomenon
- the ability to create or recreate the phenomenon

As any educator knows, evaluation can be done thoroughly or haphazardly. A *thorough* evaluation of understanding of a phenomenon would involve tests from *all four categories above*. A least-thorough evaluation would be testing only the first category; adding subsequent categories lower down will gradually increase the quality of the evaluation. In essence, evaluating understanding involves creating a *map of an understander’s ability to understand* some phenomenon—the

more we sample, and the wider, the more complete and reliable this map will become.

3.1 Prediction

Understanding of a phenomenon involves knowledge of the structure, patterns, invariants, and behaviors of that phenomenon. This means that, given state information about some of the phenomenon’s elements/variables at some point in time, it should be possible to make predictions about possible states of other elements and/or other points in time.

Here we refer to “predictions” in a very general sense. We are not restricting ourselves to predictions that are forward in time or causally downstream from the states we provide information about: if you’re provided information that the output of an AND gate was 1, then it should be possible that the inputs were both 1 as well. Predictions can occur forward, backwards or parallel in time, or even all at once.

Predictive ability can be tested using different kinds of questions. In each of these, the evaluated entity must receive some input information I , which takes the form of a set of tuples of a variable V , a time t , and the state S of that variable V at time t : $I \subset \{(V, t, S)\}$. The query set Q then has the same form, but possibly omits some values. If nothing is omitted, this results in questions of the form “will the queried variables take on the queried values at the queried times?”. Omitting the states results in questions of the form “what are possible and likely joint state-value assignments for the set of queried variables at the queried times?”. Omitting the times results in questions of the form “at what times (if ever) do you expect the queried variables to take on the queried values?”. Omitting the variables results in questions of the form “what variables can we expect to take on the queried values at the queried times?”. More complexly structured questions could be constructed by omitting different parts of each tuple in Q , or even just *partially* omitting some values (e.g. to indicate ranges of acceptable values). Sometimes, or often, multiple answers are possible, so the system should respond with an answer set A that is a collection of tuples consisting of a confidence value and an information set of the same form as Q but with all values filled in.

Constructing tests of predictive ability then involves selecting some input information I and some query set Q . The variables in I and Q must be connected through some causal chain, or else prediction of the missing values in Q from I is impossible. At least some of this causal path should overlap with the variables and relations in the phenomenon Φ under test. In fact, a proper test suite would not just randomly select I and Q to have this feature, but would attempt to get a comprehensive coverage of all variables, inward and outward relations of Φ .

It may be tempting to just select every possible input into I and every possible output into Q , so as to completely cover everything about Φ in a small number of tests. In our example, this could mean to put variables A, B and C in I and X and Y in Q . However, it would still be necessary to test a good sampling over possible input values for A, B and C, as well as times for each variable. Furthermore, failure would only suggest that *some* understanding is probably lacking, but not what part. To get a clear map of the system’s understanding of

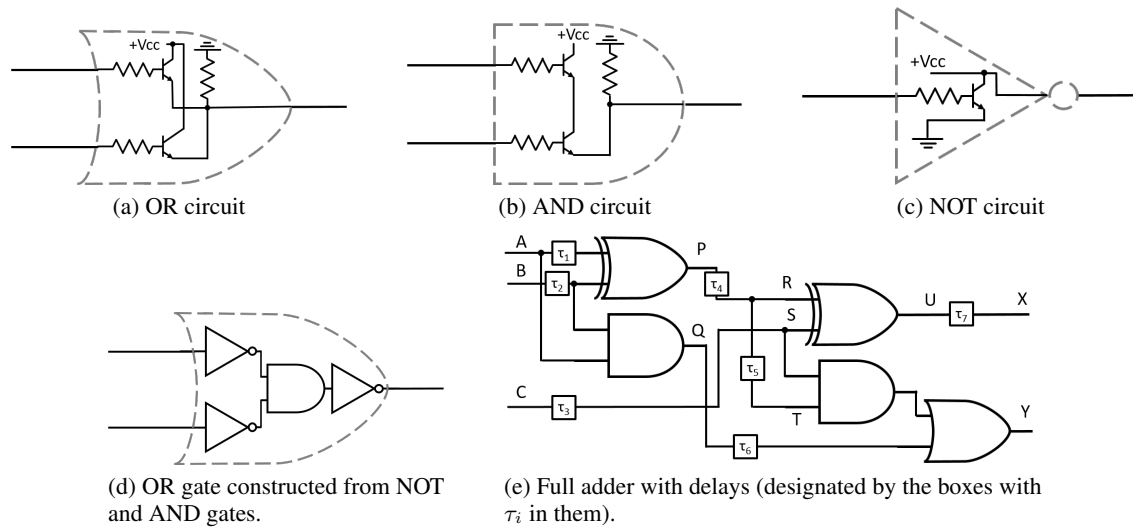


Figure 1: Examples of electronic circuits: Component view (top row) and logic view (bottom row).

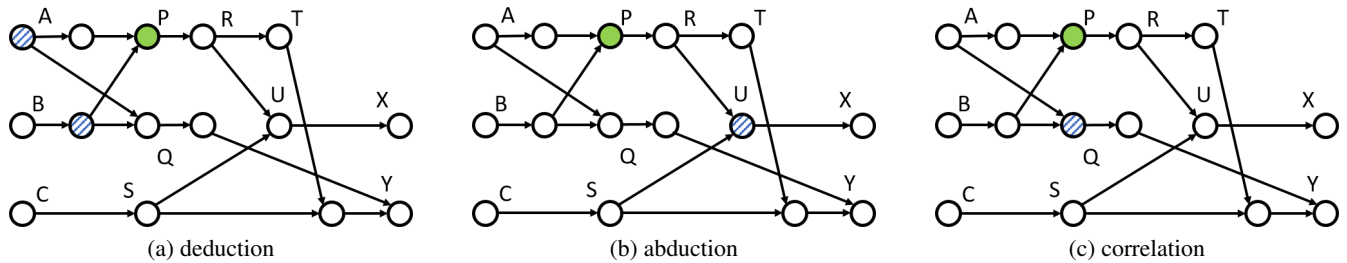


Figure 2: Variables corresponding to the adder circuit in Figure 1e including their causal relations, with different prediction tests. The green full circle represents the query variable, while the dashed blue circles represent the input information. Based on their relative positions, different modes of reasoning are necessary.

a phenomenon, a more granular and scientific approach must be adopted where understanding of different parts is gradually ascertained.

Another important consideration is how the relative locations of variables+times in I and Q affect the kind of reasoning required to produce an answer. For instance, if the inputs are upstream from the query variables, we would be testing deduction, and if they provide full coverage of the causal influences on the query variables, we expect an understanding entity to answer correctly (see Figure 2a). If the inputs are downstream from the query variables, we are testing abduction, which—unlike deduction—cannot typically guarantee the correctness of its inference given correct inputs, even if the system fully understands the related phenomena (see Figure 2b). In this case the inputs represent outcomes, that presumably constrain the possible values of the query variables. In other cases, input and query variables may occur more or less in parallel, in which case the answer can be obtained through (statistical) correlation, or through a sequence of abductions (to find a common cause) followed by deductions, or through a sequence of deductions (to find a common affected outcome variable) followed by abductions.

Success on these tests provides some evidence that understanding is present, and failure suggests that *something* went wrong. It could be that the system's understanding was flawed, and this lead to wrong answers, but it could also be that the system has very accurate models but an inability to perform abductive reasoning, or to output multiple outcomes. Such capabilities should be taken into account in the design of tests, and when failure or inaccuracies occur their source should be uncovered. Some common considerations include selecting only observable variables for I and Q , and only selecting combinations that sufficiently constrain the correct answer to a single value.

3.2 Goal Achievement

Many definitions of general intelligence refer to a system's ability to achieve complex and diverse goals in a wide variety of situations [Legg and Hutter, 2007]. This paradigm of goal achievement—albeit not always in a general, domain-independent manner—has been embraced by various AI branches like planning and reinforcement learning. Understanding should almost certainly help in the achievement of complex and diverse goals. To test this, we must assign the

system a “task” that involves the phenomenon in some way.

A task in a particular environment consists of an initial starting state (variables in the environment having particular values), a goal (a set of states with desirable joint variable assignments that must be achieved at particular times), optionally come constraints (states that result in instant failure), a body (set of observables and affordances) and some limited information about the task. For instance, in our example circuit, we might start at time $t = 0$ with all variables set to 0, the goal being to set X to value 1 before $t = 10$ under the constraint that Q must always be at 0, with the ability to affect A and B and to observe C , and knowledge of all the delays in the system. An understanding of how delays work would tell the system that if C (or U) holds some value at time t , then S (or X) will hold that same value at $t + \tau_3$ (or $t + \tau_7$), and an understanding of XOR gates will tell the system the value of U at time t given values for R and S at time t .

Goal achievement tends to involve planning, or backward chaining from the goal. A agent with an understanding of both delays and XOR gates should be able to know that in order to set X to 1 at time t , U must be 1 at time $t - \tau_7$, which means that we need $R = 1$ and $S = 0$ or $R = 0$ and $S = 1$ at time $t - \tau_7$. While an AI might not be able to output a detailed plan in a symbolic language, it might be able to demonstrate understanding by doing. A system that could do both would of course be preferable as one would have more options for how to test it. Since S is determined by C at time $t - \tau_7 - \tau_3$, it is not under the system’s control, and it must respond to what it observes the value of C to be: if it’s thought that S will be 0 at some time, A and B should be controlled to produce a 1 at R at that same time, while avoiding setting Q to 1 as well. How the task should be executed depends on the exact values of the delays, and in fact the task may be impossible to perform for an agent if τ_3 is short enough and C is controlled by an adversarial process.

A task like this could be used to evaluate understanding of the internals of the entire “adder” circuit $\mathfrak{R}_{adder}^{in}$, as well as e.g. the outward relations of XOR gates \mathfrak{R}_{XOR}^{out} (see Figure 1e).

As with the construction of prediction tests, constructing goal achievement tests requires selecting different sets of variables, and settings for them. The variables need to be selected in such a way that there is a causal path from the control variables to the goal and constraint variables that overlaps at least partially with the elements of Φ . These variables can be selected randomly, although a good test suite would need to ensure proper coverage of all elements of Φ , and ideally different combinations of elements and outward relations. An initial state can be created by starting from any realistic initial state (or some “null state”) and executing some (possibly random) behaviors on selected nodes. To determine (un)desirable state values to serve as a goal (and constraints), another (random) behavior could be executed on the nodes that the AI system will get to control from the initial state and some future state could be designated as the goal. Repeating this behavior would then be one way of solving the task, which ensures that this is in fact doable. The difficulty can be increased by executing multiple behavior sequences from the start state, and setting the goal to be a state that only oc-

curs rarely. To construct a good test—as with the other three methods discussed in this paper—good understanding of the target phenomenon Φ is necessary.

3.3 Explanation

When humans want to evaluate each other’s understanding of a phenomenon they often ask the understander to explain it to them, in part or in full. Explanation then involves summarization of the relevant elements of the phenomenon, and tracing it back to the most salient causes. This requires a capability that is currently lacking in virtually all AI systems. For this reason, initiatives like DARPA’s Explainable AI (XAI) were started [Gunning, 2016 08 10]: we don’t want our AI to just make good decisions — although this is of course important — we also want to be able to explain how they make them so that *we* can understand them.

Although having an AI that can assess its audience and come up with a good natural language summarization of the reasons for its various decisions and behaviors, this is not strictly necessary for having an AI explain itself: Explanations in special languages at a lower level of abstraction are perfectly acceptable,⁶ because ultimately we are interested in whether the system *understands the causal nature of the phenomena it observes, manages, or controls*, in a way that highlighting salient features. It is not enough here to get a printout of the system’s full knowledge base — to prove understanding the understander must be able to identify the *necessary* and *sufficient* variables and their relations to explain *precisely* what we ask it to explain — no more and no less.

In our circuit example, some actions on it will have particular effects, others will not; We ask the understander to explain why the observed results are obtained. (Again, this can be done in a descriptive language by communicating with the system, or simply by doing.) We expect an understanding system to be able to highlight/select variables that are responsible for the effects at a desired level of abstraction and detail; for circuits an explanation could be at the level of half-adders, logic gates, or electronics, depending on what the system is being asked to explain. Furthermore, we want to know what causes are *salient*: if we ask why the output of an AND gate became *true* when input A was always *true*, while input B only recently became *true*, a proper explanation would highlight input B .

Typically, when we’re asking for an explanation, we don’t want to hear “the output is a ‘dog’, because pixel (1,1) had value [255,255,0], pixel (1,2) had value [255,240,10], etc.” — while this is one form of explanation it is not the kind of necessary-and-sufficient identification of causal relations that we are after. In this example we would want something like “it’s a dog because it has I see lots of fur, two ears, two eyes,

⁶An AI system is not necessarily working with human models (our own ground truth) of phenomena, and would need to answer in terms of actions and observations that it itself can make, or in terms of its own models M_Φ . Unless the system has some method for presenting output in a reasonably human-readable way, interpretation of these models is potentially problematic, and much work is currently done on interpretability of AI systems. This topic is an interesting one but beyond our scope here.

a nose, a mouth, four legs and a leash”. And then we should be able to go deeper: “why do you think that’s an ear?”

Moving between levels of detail and abstraction when providing explanations is important as this is one way to uncover the boundaries of the understander’s model — whether the answers are e.g. derived from a set of memorized statements or whether the purported understander has a compact causal-relational model, and how far it reaches. Another way to test for boundaries is to introduce novel modifications or inputs to the task-environment, e.g. “I know you know how to play tennis, but how would you play tennis in micro-gravity?” For an understander that doesn’t understand either microgravity or tennis this becomes virtually impossible to answer in any sensible or approximate way.

3.4 (Re)creation

Creation and recreation of a phenomenon clearly involves some understanding of it. Richard Feynman famously said “What I cannot create, I do not understand”. Importantly, it should be possible to create a new instance of a phenomenon, often in a different situation (e.g. using new materials). This typically involves some understanding of what it is that makes the phenomenon what it is—i.e. what are relevant and irrelevant features.

For instance, Figure 1a and Figure 1d both show OR gates, implemented in a different way. An entity that understands NOT gates (Figure 1c), AND gates (Figure 1b) and OR gates (Figure 1a), should be able to construct another OR gate out of an AND gate and three NOT gates (Figure 1d). Of course, given these materials, it may also be a possibility to extract the lower-level electronic components from the given AND and NOT gates and use them to recreate a Figure 1a-style OR gate.

Feynman, like most physicists, wanted to understand the universe. But of course, he did not literally recreate the universe: he created models. To evaluate a system’s ability to recreate a phenomenon, we would ideally like to see and interpret the models it has formed. Here we run into a similar problem as with explanation: few if any AI systems can present their models in a way that can be understood and evaluated (by us or themselves).

To evaluate (re)creation abilities of a system, we can give it a design task, where it’s given some components / materials and the ability to combine them, and we ask that the phenomenon under test is recreated or that *something* is created that behaves a certain way. Another way is to repeatedly call for increasing levels of compression of the explanation the system gives of a phenomenon—paralleling the progress of science from Earth, Wind, Water, Fire to $E = mc^2$. Clearly, a system that can come up with a compact equation like $E = mc^2$ for some new phenomenon really truly understands that phenomenon. This is essentially what Feynman meant.

4 Acquiring Understanding

So far we have talked about evaluating the understanding of a certain phenomenon that a purported understander is to be tested on. For general AI a system’s ability to acquire understanding is most likely *more important*, in particular a system’s acquisition of arbitrary phenomena. In part, we can use

the proposed methods to evaluate this ability by taking two measurements at separate time, where the system acquires understanding in between. This would give (at a minimum) two data points along (at least) two axes of time and level(s) of understanding; whether and how these measures interact with learning methods (e.g. does the system require tutoring?) and domain (is learning speed dependent on the topic?) will affect how multi-dimensional this measure ultimately is.

We would like to know not only whether an understanding can be acquired by the system, but also how it acquires it: what information and experience is necessary, and how fast this process is. Related questions of importance are how much existing knowledge and understanding is necessary for the system to have before being able to acquire understanding of new phenomena (i.e. what is the maximum delta/time-unit for deepening/broadening understanding?), and what the knowledge transfer function looks like, i.e. in understanding new phenomena does understanding related phenomena help in the learning, and if so how much?

To evaluate this we need a more general test battery that has been discussed here, and ideally even an entire evaluation framework that allows us to easily construct different (related and unrelated) tasks in order to evaluate the understanding of a multitude of phenomena [Thórisson *et al.*, 2015 07]. Taking things even further, we would like to know how to teach different phenomena to an AI system, and know that it has received enough information to form an understanding – if it is capable of forming an understanding [Bieger, 2016 07 16].

5 Conclusions

We have discussed the need for the evaluation of understanding in intelligent agents, and what tests for this purpose might look like. Four aspects of understanding can be considered separately for this purpose: prediction, goal achievement, explanation and (re)creation. While some of these could be performed successfully by an entity that doesn’t understand, it is unlikely that such an entity could succeed on all of these metrics. By viewing phenomena and environments as hierarchical collections of variables with relations between them, we can obtain methods for generating tests for each kind of understanding. Prediction can be tested by querying the state of some variables based on some input information, goal achievement is tested by defining tasks with initial and goal states, explanation is tested by highlighting salient causal events that are most directly responsible for the explained phenomenon, and (re)creation can be tested by formulating a design task or interpreting the system’s model of the phenomenon directly.

References

- [Bieger, 2016 07 16] Jordi Bieger. Artificial Pedagogy: A Proposal. In *HLAI 2016 Doctoral Consortium*, 2016-07-16.
- [Bostrom, 2014] Nick Bostrom. *Superintelligence: Paths, Dangers, Strategies*. Oxford University Press, 2014.
- [Gunning, 2016 08 10] David Gunning. Explainable Artificial Intelligence, 2016-08-10.

- [Legg and Hutter, 2007] Shane Legg and Marcus Hutter. A collection of definitions of intelligence. In *Advances in Artificial General Intelligence: Concepts, Architectures and Algorithms*, volume 157, pages 17–24, 2007.
- [Nguyen *et al.*, 2015] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference On*, pages 427–436. IEEE, 2015.
- [Sloman, 1985] Aaron Sloman. What enables a machine to understand? In *IJCAI*, pages 995–1001, 1985.
- [Steunebrink *et al.*, 2016] Bas R. Steunebrink, Kristinn R. Thórisson, and Jürgen Schmidhuber. Growing recursive self-improvers. In *International Conference on Artificial General Intelligence*, pages 129–139. Springer, 2016.
- [Szegedy *et al.*, 2013 12 20] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. 2013-12-20.
- [Thórisson *et al.*, 2015 07] Kristinn R. Thórisson, Jordi Bieger, Stephan Schiffel, and Deon Garrett. Towards Flexible Task Environments for Comprehensive Evaluation of Artificial Intelligent Systems & Automatic Learners. In *Proceedings of AGI-15*, pages 187–196. Springer-Verlag, 2015-07.
- [Thórisson *et al.*, 2016a] Kristinn R. Thórisson, Jordi Bieger, Þrúður Thorarensen, Jóna S. Sigurðardóttir, and Bas R. Steunebrink. Why Artificial Intelligence Needs a Task Theory — And What it Might Look Like. In *Proceedings of AGI-16*. Springer-Verlag, 2016.
- [Thórisson *et al.*, 2016b] Kristinn R. Thórisson, David Kremelberg, Bas R. Steunebrink, and Eric Nivel. About understanding. In *International Conference on Artificial General Intelligence*, pages 106–117. Springer, 2016.