Turing Tests with Turing Machines

José Hernández-Orallo[1], Javier Insa-Cabrera[1], David L. Dowe[2] and Bill Hibbard[3]

[1] DSIC, Universitat Politècnica de València, Spain.
jorallo@dsic.upv.es, jinsa@dsic.upv.es
[2] Clayton School of Information Technology, Monash University, Australia.
david.dowe@monash.edu
[3] Space Science and Engineering Center, University of Wisconsin - Madison, USA.
test@ssec.wisc.edu

Abstract

Comparative tests work by finding the difference (or the absence of difference) between a reference subject and an evaluee. The Turing Test, in its standard interpretation, takes (a subset of) the human species as a reference. Motivated by recent findings and developments in the area of machine intelligence evaluation, we discuss what it would be like to have a Turing Test where the reference and the interrogator subjects are replaced by Turing Machines. This question sets the focus on several issues that are usually disregarded when dealing with the Turing Test, such as the degree of intelligence of reference and interrogator, the role of imitation (and not only prediction) in intelligence, its view from the perspective of game theory and others. Around these issues, this paper finally brings the Turing *Test* to the realm of Turing *machines*.

Keywords: Turing Test, Turing machines, intelligence, learning, imitation games, Solomonoff-Kolmogorov complexity, game theory, human unpredictability, matching pennies.

Contents

1 Introduction

The Turing Test [26] is still the most popular test for machine intelligence. However, the Turing Test, as a measurement *instrument* and not as a philosophical argument, is very different to

the instruments other disciplines use to measure intelligence in a scientific way. The Turing Test resembles a much more customary (and non-scientific) assessment, which happens when humans interview or evaluate other humans (for whatever reason, including, e.g., personnel selection, sports[1] or other competitions). The most relevant (and controversial) feature of the Turing Test is that it takes *humans* as a touchstone to which machines should be compared. In fact, the comparison is not performed by an objective criterion, but assessed by *human* judges, which is not without controversy. Another remarkable feature (and perhaps less controversial) is that the Turing Test is set on an intentionally restrictive interaction channel: a teletype conversation.

Nonetheless, there are some features about the Turing Test which make it more general than other kinds of intelligence tests. For instance, it is becoming increasingly better known that programs can do well at human IQ tests [28][27][5], because ordinary IQ tests only evaluate narrow abilities (and assume that narrow abilities accurately reflect human abilities across a broad set of tasks), which may not hold for non-human populations. The Turing Test (and some formal intelligence measures we will review in the following section) can test broad sets of tasks.

We must say that Turing cannot be blamed for all the controversy. The purpose of Turing's imitation game [33] was to show that intelligence could be assessed and recognised in a behavioural way, without the need for directly measuring or recognising some other physical or mental issues such as thinking, consciousness, etc. In Turing's view, intelligence can be just seen as a cognitive ability (or property). In fact, the standard scientific view should converge to defining intelligence as an ability that some systems: humans, non-human animals, machines —and collectives thereof—, might or might not have, or, more precisely, might have to a larger or lesser degree.

While there have been many variants and extensions of the Turing Test (see [29] or [26] for an account of these), none of them including CAPTCHAs ([34]) (and none of the approaches in psychometrics and animal cognition, either) have provided a formal, mathematical definition of what intelligence is and how it can be measured.

A different approach is based on one of the things that the Turing Test is usually criticised for: *learning*[2]. This alternative approach requires a proper definition of learning, and actual mechanisms for measuring learning ability. Interestingly, the answer to what learning is is given by notions devised from Turing machines. In the 1960s, Ray Solomonoff 'solved' the problem of induction (and the related problems of prediction and learning) [32] by the use of Turing machines. This, jointly with the theory of inductive inference given by the Minimum Message Length (MML) principle [36, 37, 35, 3], algorithmic information theory [1], Kolmogorov complexity [19, 32] and compression theory, paved the way in the 1990s for a new approach for defining and measuring intelligence based on algorithmic information theory. This approach will be summarised in the next section.

While initially there was some connection to the Turing Test, this line of research has been evolving and consolidating in the past fifteen years (or more), progressively cutting more and more links to the Turing Test. This has provided important insights into what intelligence is and how it can be measured, and has given clues to the (re-)understanding of other areas where intelligence is defined and measured, such as psychometrics and animal cognition.

[1] In many sports, to see how good a player is, we want competent judges but also appropriate team-mates and opponents. Good tournaments and competitions are largely designed so as to return (near) maximal expected information.

[2] This can be taken as further evidence for Turing's not conceiving the imitation test as an actual test for intelligence, because the issue about machines being able to learn was seen as inherent to intelligence for Turing [33, section 7], and yet the Turing Test is not especially good at detecting learning ability *during* the test.

An important milestone of this journey has been the recent realisation in this context that (social) intelligence is the ability to perform well in an environment full of other agents of similar intelligence. This is a consequence of some experiments which show that when performance is measured in environments where no other agents co-exist, some important traits of intelligence are not fully recognised. A solution for this has been formalised as the so-called Darwin-Wallace distribution of environments (or tasks) [11]. The outcome of all this is that it is increasingly an issue whether intelligence might be needed to measure intelligence. This can be interpreted or developed in at least three different ways. First, even though we might not need intelligent judges to evaluate intelligence, the use of an intelligent interrogator may be useful to conduct an *adaptive* test in a way which can be much more efficient than other simpler ways of adaptation (or no adaptation at all). Second, we may need other intelligent agents to become part of the exercises or tasks an intelligence test should contain, if we consider that intelligence has a *social* part. Third, games can be used to *compare* subjects so that intelligence scores (and levels) can be derived from this comparison, thus turning comparative tests into an alternative to single-agent tests. While we do not necessarily claim that all these three issues (intelligent adaptation, social intelligence and one-to-one comparison) will be ingredients in practical machine intelligence tests in the future, we think that it is important to study them. Also, doing this in the context of the Turing Test can be enlightening for clarifying the different ways of measuring intelligence and which of them may be more appropriate for machines.

Before undertaking this analysis, we need to recall that the Turing Test has a very relevant feature: it is an imitation game. The difficulty of imitation, however, is not directly proportional to intelligence. In fact, there might be non-intelligent Turing machines which are more difficult to imitate/identify than many intelligent Turing machines, and this difficulty seems to be related to the Kolmogorov complexity of the Turing machine. Biological intelligence is frequently biased to social environments, or at least to environments where other agents can be around eventually. This makes imitation games frequent, where the subjects to be imitated have some degree of intelligence. In fact, societies are usually built on common sense and common understanding, but in humans this might be an evolutionarily-acquired ability to imitate other humans, but not other intelligent beings in general. In addition, some neurobiological structures, such as *mirror neurons* have been found in primates and other species, which may be responsible for understanding what other animals (and people) do and will do, and for learning new skills by imitation. Nonetheless, humans frequently show remarkable levels of unpredictability. Interestingly, some of the first analyses of human predictability [30][24] linked the problem with the competitive/adversarial scenario, which is equivalent to the matching pennies problem that we will analyse in the following section, and that has been suggested as a possible way of evaluating intelligence.

The paper is organised as follows. Section 2 introduces a short account of the past fifteen (or so) years concerning definitions and tests of machine intelligence based on Turing machines, using (algorithmic) information theory and game theory. Section 3 discusses the components of a Turing Test (roles and distributions) before analysing the various possibilities of Turing Tests with Turing Machines in section 4. The use and properties (such as monotonicity) of any of these possibilities is discussed in section 5. Section 6 defines whether a recursive definition of the distributions used in the test can be used to build a hierarchy of tests where non-human intelligence references (and judges) can be found. Section 7 briefly explores how all this might develop, and touches upon concepts such as universality in Turing machines and potential intelligence, as well as some suggestions as to how machine intelligence measurement might develop in the future.

2 Machine intelligence measurement using Turing machines

There are, of course, many proposals for intelligence definitions and tests *for machines* which are not based on the Turing Test. Some of them are related to psychometrics, some others may be related to other areas of cognitive science (including animal cognition) and some others originate from artificial intelligence (e.g., some competitions running on specific tasks such as planning, robotics, games, reinforcement learning, ...). For an account of some of these, the reader can find a good survey in [20]. In this section, we will focus on approaches which use Turing machines (and hence computation) as a basic component for the definition of intelligence and the derivation of tests for machine intelligence. We will distinguish between *single-agent tests*, where the evaluee does not need to compete or interact with other agents, but just need to solve exercises or tasks (such as psychometric tests) and *adversarial tests*, where the evaluee needs to compete or interact with other agents (such as most games or the Turing test).

2.1 Single-agent tests using algorithmic information theory

Most of the views of intelligence in computer science are sustained over a notion of intelligence as a special kind of information processing. The nature of information, its actual content and the way in which patterns and structure can appear in it can only be explained in terms of algorithmic information theory. The Minimum Message Length (MML) principle [36, 37] and Solomonoff-Kolmogorov complexity [32, 19] capture the intuitive notion that there is structure – or redundancy– in data if and only if it is compressible, with the relationship between MML and (two-part) Kolmogorov complexity articulated in [37][35, chap. 2][3, sec. 6]. While Kolmogorov [19] and Chaitin [1] were more concerned with the notions of randomness and the implications of all this in mathematics and computer science, Solomonoff [32] and Wallace [36] developed the theory with the aim of explaining how learning, prediction and inductive inference work. In fact, Solomonoff is said to have 'solved' the problem of induction [32] by the use of Turing machines. He was also the first to introduce the notions of universal distribution (as the distribution of strings given by a UTM from random input) and the invariance theorem (which states that the Kolmogorov complexity of a string calculated with two different reference machines only differs by a constant which is independent of the string).

Chaitin briefly made mention in 1982 of the potential relationship between algorithmic information theory and measuring intelligence [2], but actual proposals along this line did not start until the (late) 1990s. The first proposal was precisely introduced over a Turing Test and as a response to Searle's Chinese room [31], where the subject was *forced* to learn. This *induction-enhanced* Turing Test [4] could then evaluate a general inductive ability. The importance was not that any kind of ability could be included in the Turing Test, but that this ability could be formalised in terms of MML and related ideas, such as (two-part) compression, and potentially established as a standalone single-agent test.

Independently and near-simultaneously, a new intelligence test (C-test) [12] [9] was derived as sequence prediction problems which were generated by a universal distribution [32]. The difficulty of the exercises was mathematically derived from a variant of Kolmogorov complexity, and only exercises with a certain degree of difficulty were included and weighted accordingly. These exercises were very similar to those found in some IQ tests, but here they were created from computational principles. This work 'solved' the traditional subjectivity objection of the items in IQ tests, i.e., since the continuation of each sequence was derived from its shortest explanation. However, this test only measured one cognitive ability and its presentation was

too narrow to be a general test. Consequently, these ideas were extended to other cognitive abilities in [9] by the introduction of other 'factors', and the suggestion of using interactive tasks where "rewards and penalties could be used instead", as in reinforcement learning.

Similar ideas followed relating compression and intelligence. Compression tests were proposed as a test for artificial intelligence [25], arguing that "optimal text compression is a harder problem than artificial intelligence as defined by Turing's". Nonetheless, the fact that there is a connection between compression and intelligence does not mean that intelligence can be just defined as compression ability (see, e.g., [6] for a full discussion on this).

Legg and Hutter [21] would later propose a notion which they referred to as a "universal intelligence measure" —universal because of the use of a universal distribution for the weighting over environments. The innovation was mainly their use of a reinforcement learning setting, which implicitly accounted for the abilities not only of learning and prediction, but also of planning. An interesting point for making this proposal popular was its conceptual simplicity: intelligence was just seen as average performance in a range of environments, where the environments were just selected by a universal distribution.

While innovative, Legg and Hutter's universal intelligence *measure* [21] showed several shortcomings stopping it from being a viable *test*. Some of the problems are that it requires a summation over infinitely many environments, it requires a summation over infinite time within each environment, Kolmogorov complexity is typically not computable, disproportionate weight is put on simple environments (e.g., with $1 - 2^{-7} > 99\%$ of weight put on environments of size less than 8, as also pointed out by [15]), it is (static and) not adaptive, it does not account for time or agent speed, etc.

Hernandez-Orallo and Dowe [10] re-visited this to give an intelligence *test* that does not have these abovementioned shortcomings. This was presented as an anytime universal intelligence test. The term *universal* here was used to designate that the test could be applied to any kind of subject: machine, human, non-human animal or a community of these. The term *anytime* was used to indicate that the test could evaluate any agent speed, it would adapt to the intelligence of the examinee, and that it could be interrupted at any time to give an intelligence score estimate. The longer the test runs, the more reliable the estimate.

Preliminary tests have since been done [17, 18, 22] for comparing human agents with non-human AI agents. These tests seem to succeed in bringing theory to practice quite seamlessly and are useful for comparing the abilities of systems of the same kind. However, there are some problems when comparing systems of different kind, such as human and AI algorithms, because the huge difference of both (with current state-of-the-art technology) is not clearly appreciated. One explanation for this is that (human) intelligence is the result of the adaptation to environments where the probability of other agents (of lower or similar intelligence) being around is very high. However, the probability of having another agent of even a small degree of intelligence just by the use of a universal distribution is discouragingly remote.

2.2 Adversarial tests: "Matching Pennies" and other multi-agent tests

A different, but also formal, approach to machine intelligence evaluation originates from game theory and computation, using a game known as 'matching pennies', which is a binary version of rock-paper-scissors, both of them being stateless games, since the result only depends on the current move and not on previous moves as other games. In 'matching pennies', there are two players and at each round both players signal either heads or tails. The goal of the first player, the *predictor*, is that both players' signals agree, and the goal of the second player,

the *evader*, is that the players' signals disagree. This game has been proposed as an intelligence test in the form of Adversarial Sequence Prediction [14][16] and is related to the "elusive model paradox" [3, sec. 7.5] and to the problem of human unpredictability, mentioned in the introduction. A tournament was organised in 2011 where computer algorithms can compete [http://matchingpennies.com/tournament/].

The intelligence test based on matching pennies is defined for two computational models of agent environments: Turing machines (TMs) and finite state machines (FSMs). In these tests the agent plays the predictor against an environment (TM or FSM) that plays the evader. The tests are an attempt to address several goals:

- Avoid the bias in some other formal intelligence measures [15] by testing agents against a wide set of environments.
- Provide meaningful measures for agents up to arbitrarily high levels of intelligence while assigning every agent a finite intelligence.
- Avoid assigning decreasing weight to tests against increasingly difficult environments.
- Give agents time to learn before they are measured (although in the FSM setting, the test produces a secondary measurement of agents' speed of learning).

Environments are organised into hierarchies of sets, depending on the quantity of computing resources the environments use (time for TMs, number of states for FSMs). Rather than measuring agent intelligence as a weighted sum of results against a distribution of environments, the measure is an ordinal of the highest set such that the agent can defeat every environment in the set.

Although this test measures agents against a wide set of environments, it only measures the single skill of playing the matching pennies game. Also, this test is useless for stochastic agents and environments, since any algorithm with a source of truly random bits will asymptotically win half the rounds at matching pennies.

A related approach is the notion of evaluation using multi-agent environments. The question of what the individual (or common) goals are and how the other agents are chosen is complex. In [11], the so-called Darwin-Wallace distribution is introduced where environments are generated using a universal distribution for multi-agent environments, and a number of agents (also generated by a universal distribution) populate the environment. The probability of having interesting environments and agents is very low on this first 'generation'. However, if an intelligence test is administered to this population and only those with a certain level are preserved (with high probability), we may get a second population whose agents will have a slightly higher degree of intelligence. Iterating this process we have different levels for the Darwin-Wallace distribution, where evolution is solely driven (boosted) by a fitness function which is just measured by intelligence tests.

3 Roles and distributions in Turing Tests

Some of the problems of the standard interpretation of the Turing Test are originated by the participants taking different roles: a judge, an honest (or dishonest) member of the human population and an imitator (or impostor) of the human population. It is not always clear what the goals of the three agents are and whether the three agents are really necessary for comparing humans and machines.

The basic idea behind a Turing Test is to check whether a machine has a behavioural property Υ, by comparing it to reference subject which certainly has the property. An interrogator/judge is used to try to ascertain whether the evaluee has the property.

From the Turing Test (and after the tests seen in section 2) we can distinguish the following roles:

- *Generators*: are TMs which just generate a sequential (possibly infinitely) string of 0s and 1s.
- *Interrogators*: are interactive TMs which exchange inputs and outputs and may change its outputs according to previous inputs. Typically, it is assumed that the same query is sent to all the agents. An environment can be seen as an interrogator, where the interaction is ruled by actions, observations and rewards.
- *Judges*: determines a degree of similarity between the evaluees or tells which one has the property Υ (to a higher degree). A judge has the outputs of the evaluated agents as inputs.
- *Predictors*: are TMs which try to anticipate (and output) the next bit(s) of the input.
- *Evaders*: are TMs which try to anticipate (and output) some bit(s) which do not match the next bit(s) of the input.
- *Imitators*: are special kinds of predictors which usually have a more complex input (what the agent to be imitated had as inputs and outputs) and try to have a similar behaviour.

Conceptually, generators are simple, since they do not depend on the input. Predictors and evaders are clearly more complex, since they rely on inputs and outputs. Imitators are more convoluted since they need to observeboth the inputs and outputs of other systems. Finally, interrogators and judges (and especially when these two functions come together) may need to make up models of the subjects they are interrogating and judging, and use this information to generate their outputs (queries). Nonetheless, depending on the specific setting some of these agents may become more elaborate. For instance, an imitator may try to issue outputs to get some desired behaviour from its opponent, playing a 'manipulator' or 'controller' role.

The specific roles each agent might actually take will be determined by the goals we set on each specific setting for a test, and the characteristics of the agents. Consequently, goals must be clearly determined if we want to make an unambiguous game and know exactly what it is doing.

4 Games with Turing Machines

Before giving specific instances, let us give a general definition that can be instantiated by setting the number of participants, distributions, roles and interfaces.

Definition A general Turing Test is defined as a tuple $\langle J, R, E, G_J, G_R, G_E, D_J, D_R, D_E, I\rangle$, where:

- The reference subject R is randomly chosen from a distribution D_R and follows goal G_R.
- The evaluee subject E is randomly chosen from a distribution D_E and follows goal G_E.
- The interrogator/judge J is randomly chosen from a distribution D_J and follows goal G_J.

Figure 1: Left: A two-agent unidirectional setting. Right: a two-agent bidirectional setting.

- There is an *interaction* protocol I which is executed until a condition is met (a given number of steps, a limited time or a certainty of the result).
- The test returns an assessment for E.

The evaluee E is usually given, so the distribution D_E is not necessary (or we can just express this case by saying that D_E gives all the probability to a specific individual E). An instance of the previous schema requires us to determine the distribution D_J and D_R, the goals of the three participants and, most especially, how the interaction I goes. In the classical Turing Test, we know that D_J and D_R are taken from a human population, the goal of J is to tell which of R and E has really been taken from the reference distribution D_R, the goal of R and E is to be chosen as the one who is really taken from D_R, and the interaction I is an open teletype conversation[3].

Of course, other distributions for D_R could lead to other tests, such as, e.g., a canine test, taking D_R as a dog population, and judges as other dogs which have to tell which is the member of the species or perhaps even how much intelligent it is (for whatever purpose —e.g., mating or idle curiosity).

More interestingly, we can look into possible instances using Turing machines. Let us start with instances where only two agents participate (and the judge/interrogator is replaced by a simple payoff function). After this, we will analyse the case where a judge/interrogator conducts the test.

4.1 Turing Tests by observation

If there is no interrogator which enquires or conducts the test in any way and just a judge which observes how two agents play, we have a Turing test by observation. While the capacity of telling between two agents is reduced because of the inability of conducting the test (and hence guiding the games to situations where the test can get more information), we can still explore some insightful possibilities. Figure 1 shows two cases where two agents can play the roles of generators, predictors or evaders. The reason why we use prediction or identification as a goal is the same that lies behind the proposals in section 2. We want tests that can be completely free of culture bias (or human bias), such as language, assumed previous knowledge, etc.

The setting on the left of Figure 1 is very similar to the general abstract setting which is typically used for analysing what learning is, how much complexity it has and whether it can be solved. If the generator outputs strings and the predictor has to identify what the generator does, this is clearly related to Gold's language identification in the limit [8]. If the predictor identifies the generator at some point, then it will start to score perfectly from that moment. While Gold was interested in whether this could be done in general and for every possible generator (or language), here we are interested in how well the predictor E does this on average

[3]This free teletype conversation may be problematic in many ways. Typically, the judge J wishes to steer the conversation in directions which will enable her to get (near-)maximal (expected) information (before the time-limit deadline of the test) about whether or not the evaluee subject E is or is not from D_R. One tactic for a subject which is not from D_R (and not a good imitator either) is to distract the judge J and steer the conversation in directions which will give judge J (near-) minimal (expected) information.

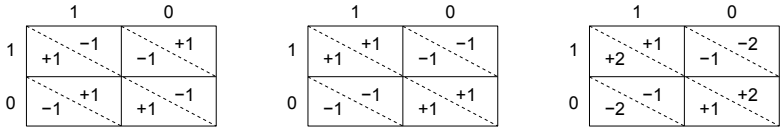

Figure 2: Payoff matrices for three different zero-sum games: matching pennies (left), pure co-ordination (middle) and battle-of-the-sexes (right).

for a randomly-chosen generator R from a distribution D_R. Also Solomonoff's setting is also very similar to this. Solomonoff proved that E could get the best estimations for R if E used a mixture of all consistent models inversely weighted by 2 to the power of their Kolmogorov complexity. While this may give the best theoretical approach for prediction and perhaps for "imitation", it does not properly "identify" R. Identification can only be properly claimed if we have one single model which is exactly as R. This distinction between one vs. multiple models is explicit in the MML principle, which usually considers just one single model, the one with the shortest two-part message encoding of said model followed by the data given this model.

We know that the higher the Kolmogorov complexity (or Levin's Kt complexity [23]) of the TM acting as a generator, the harder it will be for the predictor (the evaluee subject) to follow that. This is what many of the previous attempts for measuring intelligence using Kolmogorov complexity have attempted so far [13]. However, these attempts do not consider interaction.

The setting on the right of Figure 1 is the basic step towards interaction. A possible way of establishing a game in this setting is as follows:

Definition An observational Turing Test for Turing machines is defined as an instance of definition 4, where

- The interaction is solely given as an exchange of bits between the reference subject R and the evaluee subject E. For each step, both R and E have to output a finite binary string r_i and e_i simultaneously. Both R and E have access to the history of R and E's outputs.
- The goals of the reference and evaluee subjects, G_R and G_E, are given by a payoff matrix, which produces a reward depending on the values of r_i and e_i.
- The test returns a score as the average payoffs for agent E after a given number of interactions (or any other specified stopping criterion).

The payoff matrix is crucial for making the game informative and useful for getting some information about the behaviour of R and E. In the case where the interaction is set to binary strings of length 1, then we have a typical situation in game theory, where many special cases can be derived, as shown in Figure 2:

By changing the payoff matrix, we have many different games where we can analyse the abilities of two players[4]. In the case of matching pennies, it is pure competition what we evaluate. In the case of battle-of-the-sexes, co-ordination is important but there is some degree of selfishness. Some combinations are completely useless for evaluating agents, such as the pure co-ordination case (Figure 2, middle), where there is a pure Nash equilibrium by both agents agreeing on their outputs. In the terminology of the adversarial game (matching pennies) used in the previous section, it would be like having two predictors or two evaders.

As an evaluation test, we are precisely interested in those combinations where there is no pure or mixed Nash equilibrium. In the case of matching pennies, we have a mixed strategy

[4]Also, all these games are stateless (or static) games. More complex things take place with games for which rewards depend on previous states.

Figure 3: Left: A three-agent unidirectional setting with two predictors. Right: a two-agent unidirectional setting with one predictor and one imitator.

equilibrium when both agents make a true random choice between 0s and 1s. However, for deterministic agents without a true source of randomness the matching pennies game can be exploited as an interesting problem which can be used as a very pure and simple way of observing some prediction abilities and intelligent behaviour.

Consequently, definition 4.1 becomes useful for comparing agents as long as there is no easy equilibrium in general (or it simply does not exist), payoff can be increased by a good prediction of the opponent moves and possibly avoiding being predicted by the opponent.

The next step seems to be three-agent game.

For instance, we can just set the game with one generator and two predictors, as in Figure 3 (left). This is virtually the same case as discussed above, when we had a generator and a predictor. The problem of using a generator is that it is not adaptive and the 'queries' (outputs) chosen a priori may be useless to tell between reference and evaluee. Similarly, we could figure out a generator, a predictor and *an imitator*, as shown in Figure 3 (right) where the result of the test is given by how well the imitator resembles the predictor. While this does not go much beyond the case with a generator and two predictors, it is more convoluted, because we have many different outcomes. For instance, the generator may issue queries which can be generally captured by the predictor, and then the imitator will just behave as another predictor. In other cases, the generator may issue queries which are difficult for the predictor, then the behaviour of the predictor will be easy or difficult depending on how the predictor really is. For instance, a predictor can just behave randomly in these cases (perhaps intentionally), making it very difficult for the imitator to follow. The role of the imitator, which we will explore in more depth in the next section, is special in the sense that it may know the answers for the generator but it may answer them wrong in order to better resemble a bad predictor. The problem of this setting is that the generator has influence on the predictor, and the generator and the predictor have influence on the imitator, but the imitator has no influence on the other two agents. Consequently, it cannot properly be seen as a three-way test, since generator and imitator could be encapsulated by a single agent.

4.2 Turing Tests by interrogation

One of the distinct features of the Turing Test is that there is a special agent (originally a human) who must conduct the test with the goal of guessing who is who in the game.

So we now explore the possibilities with 'interrogators'. Interrogators can be understood in many ways and can have many different goals. In fact, the relevance of interrogators is not only that they seek to maximise information but also that the evaluee may have an effect of them, turning the evaluee into an active participant of the evaluation[5].

First, if we only consider two agents, we can just set one as interrogator and the other as

[5]Ray Solomonoff used to say that a test is also testing the tester (personal communication). In what follows, we assume that the interrogator is not the evaluee, but yet again this is related to the problem of telling between intelligent and non-intelligent agents, which might require intelligence (especially if this is meant to be done efficiently).

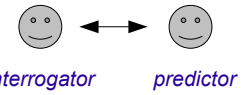

Figure 4: An adaptive test: one interrogator and a predictor.

Figure 5: Left: A three-agent bidirectional setting with two predictors. Right: a two-agent bidirectional setting with one predictor and one imitator.

the evaluee, as shown in Figure 4. This pictures an automated interrogator, trying to adapt the complexity of the exercises to the responses of the predictor is frequent in the area of *Computerised Adaptive Testing* in psychometrics. It is also one of the features of the anytime, adaptive, test in [10].

In order to explore this with three agents, a simple setting would be to have one interrogator and two predictors, as shown in Figure 5 (left). The goal of the interrogator might be to maximise the cases where one predictor is right and the other is wrong. An alternative, more detailed, option would be to determine for which kinds of problems (or difficulties), both agents differ. One possible procedure would be to adjust (increase or decrease) the Kolmogorov (or Kt) complexity of the problems in order to find the complexity where the difference between the expected accuracy of both agents is highest. This idea has some problems. First, it is not clear that the difference between the performance of the two predictors has a maximum as a function of any measure of complexity of the outputs issued by the generator. Second, there might be (infinitely) many local minima. Some of the problems are inherited from (and are well-known in) the area of adaptive testing, as in the case with one interrogator and one predictor, already mentioned above.

A next step, following the standard interpretation of the Turing Test, would be to introduce the figure of an 'imitator'. The gist of this role is linked to the existence of a good reference, since an imitator could look stupid in order to imitate a less skilled reference ability, such as a computer program having to slow down when multiplying two 5-digit numbers together in order to impersonate a human. As already mentioned, while there is a dependency, the difficulty of imitation is not proportional to the intelligence of the imitated subject. However, we are especially interested (as Turing was) in the ability of imitating *intelligent* agents, since this is a sufficient (but not necessary) condition for intelligence. A way out of this situation is to figure out a setting where 'purposed' evaluees need to imitate agents that are forced to act intelligently (or, at least, to predict well). A setting which includes an imitation role in this way is shown in Figure 5 (right).Here we have three players: a reference subject, which plays the predictor role, and an evaluee subject, which plays the imitator role.

Definition A Turing Test with Turing machines as imitator and interrogator is defined as an instance of definition 4, where

- The interaction goes as follows. A finite string (query) is output by interrogator J, then R and E must output a string simultaneously, then J issues another query, and so on. The three agents 'see' what the other agents have output so far.
- The goal of the interrogator, G_J, is defined as a reward function which directly depends on how similar the outputs of J and R are, and how dissimilar the outputs of R and E

are. These two components of the reward function could have a weight α, where $\alpha = 1$ means that only the similarity between the outputs of J and R is used and $\alpha = 0$ means that only the dissimilarity between the outputs of R and E are considered.

- the goal of the reference subject, G_R is defined as a function which depends on how similar the outputs of J and R are. As we can see, the reward functions for J and R are 'compatible' (we can improve both at the same time).
- the goal of the evaluee subject, G_E is defined as a function of the similarity between R and E. Note that this could be incompatible with part of the reward for the interrogator.
- The test returns a score as the average payoffs for agent E after a given number of interactions (or any other specified stopping criterion).

It is not clear which value of α is most reasonable. It seems clear that it cannot be 1, because in that case the interrogator would just make very simple questions and all the rewards would be compatible, leading to a (Nash) equilibrium. A value of $\alpha = 0$ is possible, but it may lead to J outputting very difficult sequences (e.g. random), so R would be completely lost and would answer almost randomly, whereupon E would have many problems to imitate R.

As we see, the three players have different characteristics and different reward systems. The game is also asymmetric, and the interrogator knows who the reference is and who the evaluee is. In order to make it symmetric, the test could be defined as a series of executions of definition 3, where the roles of R and E are exchanged.

Again, as in the predictor-evader case (e.g., matching pennies), whenever we have interaction the output has influence on future inputs, so agents need to determine the best outputs for future rewards. This means that in some occasions some agents may act suboptimally (non-rationally in the terminology of game theory) for a local decision in order to maximise global rewards. This is exactly what happens in reinforcement learning. In fact, we could say that being a generator is not demanding at all, being a good predictor is demanding as we know from passive, sequential prediction in Solomonoff's sense, being a good imitator is more demanding since it sets the setting in a kind of input-output query structure where the agent does not have influence on the inputs (so it is not query learning but interactive learning, as, e.g., reinforcement learning) and finally, the interrogator is the most demanding of all.

In this three-agent game we have many possibilities. If the predictor is able to predict well and the imitator follows the predictor, then the reward for the interrogator will not be optimal (depending on α). If the predictor performs well and the imitator cannot cope with the predictor the rewards for the interrogator will be high. If the predictor is not able to predict well, and the imitator does not follow the predictor the reward for the interrogator will not be optimal (depending on α). Reducing the difficulty of the queries is a possible solution. Finally, if the predictor is not able to predict well and the imitator follows the predictor, then the interrogator will have to reduce the difficulty of the queries, but this may not improve the situation for the interrogator. However, any of the three agents can take suboptimal actions in order to maximise the overal reward (for this it is relevant whether the agents know what the stopping criterion for the test is).

5 Monotonicity, total orders and intelligence scales

As we have seen, not many settings of the Turing test work for Turing machines, especially because some combinations lead to easy equilibrium, evaluation is inefficient or imitation does

not have a purposed reference. The previous instances also make it more explicit that the distribution D_R over the agents that the evaluee has to imitate or compete with is crucial. And the way in which the interrogator is chosen (D_J) is also important for the efficiency of the test.

Let us first explore one of the cases where things seem better delimited: the matching pennies case. One should expect that if an agent a is more intelligent than b then the comparison would give a better score to a than b. From here, especially if we want to create a scale from pairwise comparisons, we would like to have the transitivity property, i.e., using $<$ as an order relation, we would like that for every a, b and c we have that $a < b$ and $b < c$ implies $a < c$. We see that this is not the case, as the following example shows:

Example: *Let us consider an agent a_1 which always outputs 1. Consider a second agent a_2 which starts with a 0 and then switches to 1 (forever) if the opponent started with 1 or stays with 0 (forever) if the opponent started with 0. Finally, a third agent a_3 starts with a 0 and then switches to 1 (forever) if the opponent started with 0 or stays with 0 if the opponent started with 1. The game is always between a predictor and an evader, and the previous description is given for the role of predictor. When they play as evaders, the output bits are just the opposite. For instance, if a_1 plays as predictor and a_2 as evader, the outputs would be* 11111... *for a_1 and* 10000... *(the opposite of* 01111...*) for a_2. Clearly, the outputs of a_1 and a_2 always differ from the second bit and hence a_2, which is playing the evader, wins from that moment on. If a_1 plays as evader and a_2 as predictor, the outputs would be* 00000... *(the opposite of* 11111...*) for a_1 and* 00000... *for a_2. Clearly, the outputs of a_1 and a_2 always match from the second bit and hence a_2, which is playing the predictor, wins from that moment on. Following the same procedure for the rest of cases, it is easy to see that right from the second bit, the relation with expected rewards (averaging games with opponents playing both roles) are $\mathbb{E}(\sum r(a_1)) < \mathbb{E}(\sum r(a_2))$, $\mathbb{E}(\sum r(a_2)) = \mathbb{E}(\sum r(a_3))$ but $\mathbb{E}(\sum r(a_3)) < \mathbb{E}(\sum r(a_1))$.*

This clearly shows a lack of monotonicity even for deterministic agents.

While this excludes the possibility of finding a complete order, we may still find partial orders. For instance, in [14], a hierarchy of agents is constructed by using some proper definition of finite intelligence (by using finite states or finite time). This work specifically shows that every agent in a certain class has a finite intelligence and that there are agents at every level of intelligence. In order to belong to a class at a level of intelligence, an agent has to defeat (learn to predict) all agents of a given complexity. So here we can induce a partial order, where we can compare all elements at different classes (in the hierarchy). Inside each class we may just solve the issue by just putting all the elements in the same equivalence class and say that they are of comparable intelligence. For instance, the three machines above would clearly be at the lowest class.

Nonetheless, this hierarchy, in practice, may be misleading. Imagine an agent a_1 which is able to defeat all agents of class $\leq n$, and an agent a_2 which is able to defeat all agents of class $\leq m$, with $n << m$. Now imagine a third agent a_3 which is exactly equal as a_2 but that whenever it identifies a specific third agent a_4 in class 0, it generates an output in such a way that it loses. With this simple scenario, we would have to put a_3 in class 0, even though on average it is much better than a_1 (in class n) and almost identical to a_2 (in class m). With a similar rationale, we could put humans in class 0 as well. The previous rationale can be extended to any of the other complex settings in previous section.

All this gives support to the notion of average results as a more meaningful way of deriving a scale or measure for agents. This is what many game and sports competitions do, by arranging pairings and averaging (or summing results), such as a league or other hierarchical methods. But any averaging method over an infinite set (of Turing machines) *requires* a distribution.

A first idea might be a universal distribution (or computable approximations based on, e.g.,

Levin's Kt). A universal distribution would lead to assigning more intelligence to those agents which are very good at identifying and beating very simple opponents (while still losing against more complex opponents).

Putting some order (literally) here requires a process where we can start from very simple agents but eventually we can build some structure here. One easy option is to use some well-established reference, as the standard Turing Test does, by setting humans as reference. However, as discussed elsewhere (see, e.g., [9]), this is useless below and beyond this level. An alternative approach which does not require any arbitrary reference leads to the recursive proposal in the following section.

6 Recursive Turing Tests for Turing machines

The idea of recursive populations where higher intelligence levels have been selected as an application of a test can be linked to the notion of *recursive Turing Test* [28, sec. 5.1], where the agents which have succeeded at lower levels could be used to be compared at higher levels. However, there are many interpretations of this informal notion of a recursive Turing Test. The fundamental idea is to eliminate the human reference from the test using recursion, either as the subject that has to be imitated or the judge/interrogator which is used to tell between the subjects. We need to define an agent distribution D_R where we can sample from.

Let us start with the case where we only have two agents (reference and evaluee) playing the matching pennies game (see definition 4.1).

Definition The distribution D_R can be recursively obtained as follows:

Set M_0 equal to a startup probability measure over TMs, (e.g. a universal distribution)
Set $i = 0$
repeat
  Generate a set S_E of machines from M_i
  forall the $E_j \in S_E$ **do**
    Generate a set S_R of machines from M_i
    forall the $R_k \in S_R$ **do**
      Apply the test in definition 4.1 with the reference machine R_k to the evaluee E_j
    end
    E_j's intelligence at degree i, denoted by $\Upsilon_i(E_j)$ is just averaged from this set of tests
  end
  Set $i = i + 1$
  Calculate a new distribution M_i for each agent $E_j \in S_E$:

$$M_i(E_j) = \frac{rM_{i-1}(E_j) + (1-r)\Upsilon_{i-1}(E_j)}{\sum_{E_l \in S_E} rM_{i-1}(E_l) + (1-r)\Upsilon_{i-1}(E_l)}$$

  where r is a rate parameter
until *StoppingCriterion*;

When the *StoppingCriterion* is met (e.g. a given value for i), the distribution D_R is calculated from the probability measure M_i.

The outcome and properties of the previous algorithm depend on many issues. For instance, the first set taken from the distribution S_E can be populated until $M(S_E) > c$ with c sufficiently close to 1 (this means taking a large proportion of the probability mass). The size of the second set can also be important as well as how much time (or steps) are left for each test. Similarly, the rate parameter r can be used to modify the distibution with a different pace.

The previous definition is clearly related to the field of evolutionary game theory [38]. Nonetheless, strictly speaking, we do not have replicator equations and we do not try to build any agent, just to categorise them. We just define a distribution for Turing machines with the goal of having a sequence of distributions (or measures) M_i which give higher probability to agents with higher intelligence as long as i is greater.

Even if reasonable samples, heuristics and step limitations are used, the previous approach seems intractable. A better approach to the problem would be some kind of propagation system, such as Elo's rating system of chess [7], which has already been suggested in some works and competitions in artificial intelligence. A combination of a *soft* universal distribution, where simple agents would have slightly higher probability, and a one-vs-one credit propagation system such as Elo's rating (or any other mechanism which returns maximal expected information with a minimum of pairings), could feasibly aim at having a reasonably good estimate of the relative abilities of a big population of Turing machines, including some AI algorithms amongst them. With more and more actual machines and AI algorithms being assessed, non-anthropocentric Turing Tests could be implemented using these (distributions of) machines.

The discussion on recursive Turing Tests has focussed on the adversarial version of the game as in definition 4.1. However, the role of the interrogator is fundamental to make the test more effective and more efficient, so it is then advantageous to use interrogators with high intelligence. Instead of 'evolving' another distribution for interrogators separately, a better idea may be to use general agents (e.g., reinforcement learning agents) which can take any of the roles we have discussed in this paper: predictor/evader, imitator and interrogator, and let a same recursion algorithm work for them, by exchanging roles for each iteration.

7 Final Remarks

An imitation game is based on a reference subject (or population) which clearly has a property, and the test is conducted in such a way that the evaluee has to show that it can also have the property, by imitating the reference (in general behaviour, and not only for the property). While this is a much too general approach for many properties, it makes sense for intelligence, since imitation and prediction are closely related, and both are ultimately related to learning and intelligence. In this context, this paper has investigated the expression of Turing Tests where references and interrogators are Turing machines.

This view of the (recursive) Turing test in terms of Turing machines has allowed us to connect the Turing test with fundamental issues in computer science and artificial intelligence, such as the problem of learning (as identification), Solomonoff's theory of prediction, the MML principle, game theory, etc. These connections go beyond to other disciplines such as (neuro-)biology, where the role of imitation and adversarial prediction are fundamental, such as predator-prey games, mirror neurons, common coding theory, etc. In addition, this has shown that the line of research with intelligence tests derived from algorithmic information theory, some games such as matching pennies, and the recent Darwin-Wallace distribution are also closely related to this as well. This (again) links this line of research to the Turing test, where humans have been replaced by Turing machines.

This sets up many avenues for research and discussion. For instance, the idea that the

ability of imitating relates to intelligence can be understood in terms of the universality of a Turing machine, i.e. the ability of a Turing machine to emulate another. If a machine can emulate another, it can acquire all the properties of the latter, including intelligence. Imitation is then a sufficient condition for intelligence. The difference between a UTM and a, let us say, a universal imitating TM is that the latter 'learns' to behave like any other machine, and it is not programmed to do so. This is at the core of Turing's discussion about learning machines in section 7 of the very same paper [33] where he introduced the Turing Test.

Acknowledgements

This work was supported by the MEC projects EXPLORA-INGENIO TIN 2009-06078-E, CONSOLIDER-INGENIO 26706 and TIN 2010-21062-C02-02, and GVA project PROMETEO/2008/051. Javier Insa-Cabrera was sponsored by Spanish MEC-FPU grant AP2010-4389.

References

[1] G. J. Chaitin. On the length of programs for computing finite sequences. *Journal of the Association for Computing Machinery*, 13:547–569, 1966.

[2] G. J. Chaitin. Godel's theorem and information. *International Journal of Theoretical Physics*, 21(12):941–954, 1982.

[3] D. L. Dowe. MML, hybrid Bayesian network graphical models, statistical consistency, invariance and uniqueness. In P. S. Bandyopadhyay and M. R. Forster, editor, *Handbook of the Philosophy of Science - Volume 7: Philosophy of Statistics*, pages 901–982. Elsevier, 2011.

[4] D. L. Dowe and A. R. Hajek. A non-behavioural, computational extension to the Turing Test. In *Intl. Conf. on Computational Intelligence & multimedia applications (ICCIMA'98), Gippsland, Australia*, pages 101–106, February 1998.

[5] D. L. Dowe and J. Hernandez-Orallo. IQ tests are not for machines, yet. *Intelligence*, 40(2):77–81, 2012.

[6] D. L. Dowe, J. Hernández-Orallo, and P. K. Das. Compression and intelligence: social environments and communication. In J. Schmidhuber, K.R. Thórisson, and M. Looks, editors, *Artificial General Intelligence*, volume 6830, pages 204–211. LNAI series, Springer, 2011.

[7] A. E. Elo. *The rating of chessplayers, past and present*, volume 3. Batsford London, 1978.

[8] E. M. Gold. Language identification in the limit. *Information and control*, 10(5):447–474, 1967.

[9] J. Hernández-Orallo. Beyond the Turing test. *Journal of Logic, Language and Information*, 9(4):447–466, 2000.

[10] J. Hernández-Orallo and D. L. Dowe. Measuring universal intelligence: Towards an anytime intelligence test. *Artificial Intelligence Journal*, 174:1508–1539, 2010.

[11] J. Hernández-Orallo, D. L. Dowe, S. España-Cubillo, M. V. Hernández-Lloreda, and J. Insa-Cabrera. On more realistic environment distributions for defining, evaluating and developing intelligence. In J. Schmidhuber, K.R. Thórisson, and M. Looks, editors, *Artificial General Intelligence*, volume 6830, pages 82–91. LNAI, Springer, 2011.

[12] J. Hernández-Orallo and N. Minaya-Collado. A formal definition of intelligence based on an intensional variant of Kolmogorov complexity. In *Proc. Intl Symposium of Engineering of Intelligent Systems (EIS'98)*, pages 146–163. ICSC Press, 1998.

[13] José Hernandez-Orallo and N. Minaya-Collado. A formal definition of intelligence based on an intensional variant of Kolmogorov complexity. In *Proceedings of the International Symposium of Engineering of Intelligent Systems, ICSC Press*, pages 146–163, 1998.

[14] B. Hibbard. Adversarial sequence prediction. In *Proceeding of the 2008 conference on Artificial General Intelligence 2008: Proceedings of the First AGI Conference*, pages 399–403. IOS Press, 2008.

[15] B. Hibbard. Bias and no free lunch in formal measures of intelligence. *Journal of Artificial General Intelligence*, 1(1):54–61, 2009.

[16] B. Hibbard. Measuring agent intelligence via hierarchies of environments. *Artificial General Intelligence*, pages 303–308, 2011.

[17] J. Insa-Cabrera, D. L. Dowe, S. España, M. V. Hernández-Lloreda, and J. Hernández-Orallo. Comparing humans and ai agents. In *AGI: 4th Conference on Artificial General Intelligence - Lecture Notes in Artificial Intelligence (LNAI)*, volume 6830, pages 122–132. Springer, 2011.

[18] J. Insa-Cabrera, D. L. Dowe, and J. Hernández-Orallo. Evaluating a reinforcement learning algorithm with a general intelligence test. In *CAEPIA - Lecture Notes in Artificial Intelligence (LNAI)*, volume 7023, pages 1–11. Springer, 2011.

[19] A. N. Kolmogorov. Three approaches to the quantitative definition of information. *Problems of Information Transmission*, 1:4–7, 1965.

[20] S. Legg and M. Hutter. Tests of machine intelligence. In *50 years of artificial intelligence*, pages 232–242. Springer-Verlag, 2007.

[21] S. Legg and M. Hutter. Universal intelligence: A definition of machine intelligence. *Minds and Machines*, 17(4):391–444, November 2007.

[22] S. Legg and J. Veness. An Approximation of the Universal Intelligence Measure. In *Proceedings of Solomonoff 85th memorial conference*. Springer, 2012.

[23] L. A. Levin. Universal sequential search problems. *Problems of Information Transmission*, 9(3):265–266, 1973.

[24] D. K. Lewis and J. Shelby-Richardson. Scriven on human unpredictability. *Philosophical Studies: An International Journal for Philosophy in the Analytic Tradition*, 17(5):69 – 74, October 1966.

[25] M. V. Mahoney. Text compression as a test for artificial intelligence. In *Proceedings of the National Conference on Artificial Intelligence, AAAI*, pages 970–970, 1999.

[26] G. Oppy and D. L. Dowe. The Turing Test. In Edward N. Zalta, editor, *Stanford Encyclopedia of Philosophy*. Stanford University, 2011. http://plato.stanford.edu/entries/turing-test/.

[27] P. E. Ruiz. Building and solving odd-one-out classification problems: A systematic approach. *Intelligence*, 39(5):342 – 350, 2011.

[28] P. Sanghi and D. L. Dowe. A computer program capable of passing IQ tests. In *4th Intl. Conf. on Cognitive Science (ICCS'03), Sydney*, pages 570–575, 2003.

[29] A. P. Saygin, I. Cicekli, and V. Akman. Turing test: 50 years later. *Minds and Machines*, 10(4):463–518, 2000.

[30] M. Scriven. An essential unpredictability in human behavior. In B. B. Wolman and E. Nagel, editors, *Scientific Psychology: Principles and Approaches*, pages 411–425. Basic Books (Perseus Books), 1965.

[31] J. R. Searle. Minds, brains and programs. *Behavioural and Brain Sciences*, 3:417–457, 1980.

[32] R. J. Solomonoff. A formal theory of inductive inference. *Information and Control*, 7:1–22, 224–254, 1964.

[33] A. M. Turing. Computing machinery and intelligence. *Mind*, 59:433–460, 1950.

[34] L. von Ahn, M. Blum, and J. Langford. Telling humans and computers apart automatically. *Communications of the ACM*, 47(2):56–60, 2004.

[35] C. S. Wallace. *Statistical and Inductive Inference by Minimum Message Length*. Information Science and Statistics. Springer Verlag, May 2005. ISBN 0-387-23795X.

[36] C. S. Wallace and D. M. Boulton. An information measure for classification. *Computer Journal*, 11(2):185–194, 1968.

[37] C. S. Wallace and D. L. Dowe. Minimum message length and Kolmogorov complexity. *Computer Journal*, 42(4):270–283, 1999.

[38] J. W. Weibull. *Evolutionary game theory*. The MIT press, 1997.