

Michael Reid

PWR 1GMR-02

Compressionism: A New Theory of the Mind Based on Data

Compression

Since the invention of computers, computer scientists have aspired to create intelligent machines, a goal known as strong artificial intelligence (AI). The philosophy of artificial intelligence attempts to determine what kind of intelligence can be achieved by computers. There are two questions at stake: can a computer have intellectual ability as humans do, and can a computer have consciousness as humans do? If a computer can achieve the same intelligence (in both ability and consciousness) that humans have, it is said to have a mind. Much progress in the philosophy of the mind has been based on AI. Behaviorists such as Alan Turing and functionalists such as Hilary Putnam have developed theories of the mind that allow certain computers to be considered intelligent. Other writers, such as John Searle, have given strong challenges to the possibility of computer intelligence that have not been adequately addressed. Recently, approaches to AI based on a branch of theoretical computer science, known as algorithmic information theory, have contributed to this subject by proposing a computational view of induction and understanding. In this paper, I will summarize these ideas and then show how algorithmic information theory leads to compressionism, a theory of the mind that views intelligent beings as data compression systems. I will clarify the ideas behind it and show how they relate to other ideas, especially Searle's Chinese room argument.

Foundations of AI: What is a Computer?

In 1950, Alan Turing published “Computational Machinery and Intelligence”, an article laying

the foundations for artificial intelligence and arguing that strong AI is possible. He begins by considering the question “Can machines think?” but rejects it due to the vagueness of the words “machine” and “think”. In order to formulate a better version of the question, Turing defines the machines in question as digital computers, and clarifies a digital computer to be a system that does the same operations that a human computer could do (Turing, “Computing Machinery” 3). The human computer is a system in which a person is given a book of unambiguous rules to follow (with no authority to improvise or deviate from the rules) and unlimited paper and pencil to use for his calculations. When we use a straightforward procedure (algorithm) such as long division to perform calculations, we play the part of the human computer. Though this definition of a digital computer in terms of a human computer seems odd, it is a reasonable definition because it corresponds to the simple, unambiguous operations that digital computers can do. Turing shows that all digital computers, regardless of their architecture and physical realization, are equivalent in the terms of the calculations they can do and differ only in the time and memory required to perform the calculations, so we can fix our attention to a single programmable machine and think of how to write an intelligent program for it (Turing 7).

Turing formally defines such a machine in his paper “On Computable Numbers, with an Application to the Entscheidungsproblem”. This machine (now called a Turing machine) can be programmed by specifying instructions in a table (Turing, “On Computable Numbers” 231). It can receive input, perform computations using unlimited memory, and give output. Turing posits that a Turing machine can execute any algorithm, the kind that a human computer could perform (Turing 249). This idea is known as the Church-Turing thesis (Russel 8). Some support for the Church-Turing thesis can be offered in that programming languages such as Python are equivalent to Turing machines (though they are much more usable and efficient). We say that both Python and Turing machines are Turing-complete and so it doesn't especially matter which one we use as a model for computation.

Turing's definition of a digital computer is generally accepted as a theoretical model for a digital computer (Russel 8), but in practice the speed and memory of a machine are very important. However, these concerns are irrelevant to the question of whether an intelligent digital machine is *imaginable*, which must be established before determining whether they are possible in practice.

The Turing Test and Behaviorism

With his definition of a digital computer, Turing replaces the question “Can machines think?” with “Are there imaginable digital computers which would do well in the imitation game?” (Turing, “Computational Machinery” 7). In the imitation game (now known as the Turing test), a computer has a text-based conversation with a human judge who does not know if he or she is talking to a computer or to a human. According to Turing, if the judge guesses that the machine is a human as often as he or she guesses that the actual human is a human, then the machine must be intelligent and conscious (Turing 1-2). The conversation is text-based because a textual conversation is enough to demonstrate intelligent conversational skills in any possible subject that could be brought up (Turing 2). This test embodies behaviorism¹, the idea that we can tell if a system is intelligent from its behavior. Turing never claims that the Turing test is a definition of intelligence, nor that every intelligent entity must be able to pass the Turing test; he only claims that, whatever intelligence is, a machine that passes the Turing test must have it.

Behaviorists such as Turing are motivated by practical considerations. In his proposal of the Turing test, Turing meant to allow AI researchers to focus on creating computers with intelligent behavior instead of worrying about whether the intelligent-seeming computers they were trying to create were actually intelligent and conscious. Instead of predicting when strong AI will be achieved,

¹ I'm only using “behaviorism” to refer to the idea that behavior identical to that of intelligent beings implies intelligence, not any other idea that is associated with the word behaviorism.

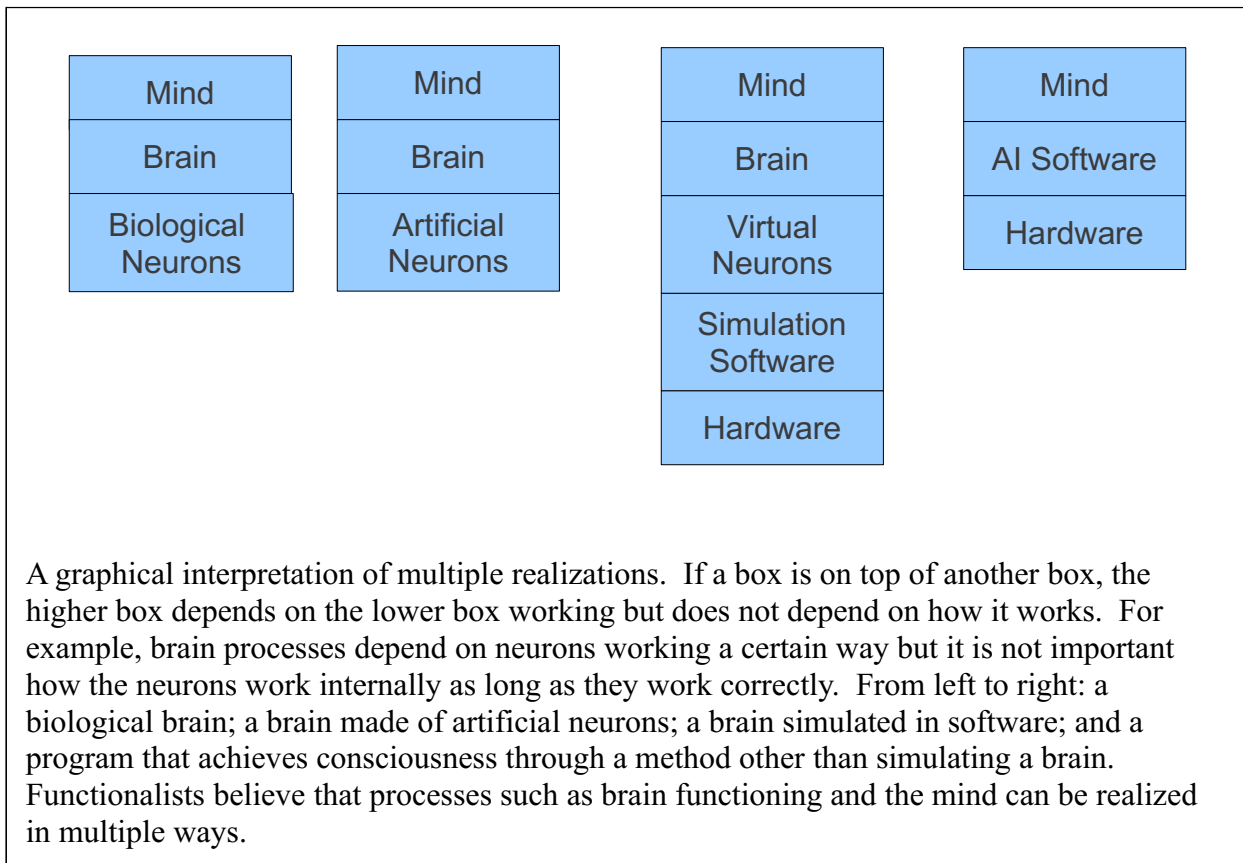
Turing attempts to predict when the Turing test will be passed² because it is a more clearly defined question that Turing considers mostly equivalent (Turing 8). The Turing test is easy to give and, in many cases, intelligent behavior is all we really care about. Programmers generally only worry about whether their programs solve the problem, not whether they are “actually” intelligent, and acceptance of the Turing test removes this distinction. Some evidence for the Turing test can be offered in that passing it requires many capabilities, such as natural language processing, knowledge representation, automated reasoning, and machine learning (Russel 2-3). In practice AI researchers try to create programs that have these capabilities instead of jumping directly to trying to pass the Turing test (Russel 3). Still, Turing's idea that we can tell if a system is intelligent in a particular area by simply testing it lives on.

Functionalism: Seeing the Big Picture

What would happen if, gradually, every neuron in your brain were replaced with an equivalent artificial electronic device³? Your behavior would certainly not change: by definition, the artificial neurons have the exact same behavior as biological neurons and the entire brain would produce the same behavior given the same sensory input. But would you still be conscious? What if the artificial neurons were replaced with virtual neurons in a running computer program that simulates a human brain? A functionalist would say that all of these situations are equivalent because they are *functionally isomorphic*: there is an obvious correspondence between biological neurons and artificial neurons or

2 His predictions turned out to be overly optimistic, mostly because he only took computer memory into account, not the speed of computations.

3 It is not clear whether this is possible. It depends on the strong Church-Turing thesis, which posits that any physical process can be accurately simulated by an algorithm or Turing machine (possibly with a random component). Writers such as Roger Penrose believe that some processes in the brain cannot be simulated accurately by an algorithm. I will assume the strong Church-Turing thesis because relatively few AI writers deny it. Note that quantum computing does not disprove the strong Church-Turing thesis: it is extremely inefficient but still possible for an ordinary Turing machine to simulate a quantum computer, and the strong Church-Turing thesis makes no claims about computation time. A rejection of the strong Church-Turing thesis would make many of my conclusions invalid.



virtual neurons. Therefore, all of the different systems (the biological brain, the artificial brain, and the virtual brain) would have the same consciousness.

Functionalism is a theory of the mind that allows a mental function (such as the function of a neuron) to be implemented in multiple ways. We should separately explain how neurons work as they do and how the brain works based on the behavior of neurons. A consequence of this is that two functionally isomorphic systems must be equivalent. Hilary Putnam, who developed many of the ideas behind functionalism, explains that “Two systems are functionally isomorphic if there is a correspondence between the states of one and the states of the other that preserves functional relations” (Putnam 291). This definition seems somewhat abstract but can be clarified through example. Consider an on/off switch on one side and a coin on the other. We can see that the two systems both have two states each (on and off, or heads and tails). We can also see that functional relations are

preserved: if the system is not acted on, it will remain in its state, and if it is flipped, it will switch to the other one. Thus we would say that the light switch and the coin are *functionally isomorphic* when viewed in this way. As a more practical example, consider an analog versus a digital watch. Both systems store the current time of day, the former through the position of gears and the latter through a digital electronic system. Both transition between states in the same way, namely by advancing to the state representing the next second as a second passes. We would say that the two watches are functionally isomorphic and, if the user does not care about the difference in how the time is displayed, completely equivalent. Importantly, we don't care *how* the watches store the current time; we just care that it stores the time somehow and advances the time every second. It is the high-level features (storing the time and advancing it) of the watches' functioning that make them watches, not how these features are realized.

This is an example of the functionalist idea of multiple realizations. According to this idea, a function (such as time-keeping, the function of a neuron, facial recognition, or general intelligence) can be implemented equivalently in multiple ways, and it is the function itself that is important, not its realization. Putnam explains this idea: “it doesn't matter at all that the physical realization of those states are totally different. So a computer made of electrical components can be isomorphic to one made of cogs and wheels or to human clerks using paper and pencil ” (Putnam 293). This idea makes intuitive sense. In everyday life, we are concerned mostly with the properties of an object (such as shape, appearance, mass, or the ability to tell time), not the low-level makeup of the object that causes it to have these properties. Functionalism would extend this reasoning to the mind: it is the mental processes that take place in brains that make them minds, not the physical processes that cause these mental processes. It shouldn't matter if our neurons are analog or digital any more than it should matter for watches: both create the same mental processes. We could even replace part of our brain (say, the part that detects lines and other low-level features in our vision) with a computer that performs the

same function differently without affecting our mental functioning⁴. Intelligence and consciousness must emerge from brains made of equivalent components just as they do from biological brains.

Along these lines, Putnam also argues that high-level phenomena can only be explained effectively through high-level explanations. For example, if one must explain why a square peg will fit into a square hole in a board but not a circular hole, then the appropriate explanation is one in terms of geometry and rigidity, not one in terms of the molecules in the peg and board and their particle dynamics (Putnam 295-296). Even if the latter explains the phenomenon, it is so complex as to be useless and we should be concerned primarily with high-level explanations for high-level phenomena. So if we want to understand intelligence, we should look at its high-level properties (such as the abilities it gives to those who possess it) rather than studying only neurons. We can't say that we understand intelligence or the brain if we only understand how neurons work; we have to look at the high-level system implemented by the neurons or other mechanism. In the extreme, perhaps there is only one function (intelligence) which the brain approximately implements and which can be implemented equivalently in completely different ways.

The Chinese Room and Semantics

Behaviorists and functionalists generally accept computationalism: they believe that it is theoretically possible for a digital computer to be an intelligent mind. The most obvious such computer would simulate every neuron in the human brain and the interactions between them. This machine would require more computational resources than are currently available in practice today but is theoretically possible. John Searle is a prominent critic of computationalism and believes that such a simulation might be possible but would not constitute a mind.

4 A functionalist might not want to extend this to the entire brain: otherwise, they would have to accept behaviorism. But maybe functionalists can accept that, as long as there is no consciousness within a part of a system, we can replace it with an part that behaves the same way.

He presents the well-known Chinese room argument to argue against behaviorism and functionalism. The argument proposes a thought experiment in which Searle (who knows English but not Chinese) plays the part of the human in the human computer defined by Turing. Searle is confined to a room and has a book of English instructions on how to manipulate Chinese characters using deterministic rules (Searle, "Minds" 3). People outside the room write questions in Chinese and give them to Searle, who manipulates the text according to the rules he is given to produce a response, which he hands back. These rules are chosen so that Searle can carry on an intelligent conversation in Chinese (passing the Chinese Turing test). However, Searle claims that, despite this, he does not understand Chinese and all the symbols are meaningless to him (Searle 4). More broadly, Searle asserts that the system of him, the rules, and the symbols does not understand Chinese. Searle suggests that he could commit the rules and symbols to memory and thus have the entire system be inside his head, still without understanding Chinese (Searle 5). The Chinese room represents a computer: Searle and the instruction book represents the CPU and the symbols written on paper (or alternatively, memorized by Searle) represent the computer's memory. By analogy, Searle asserts that a computer simulation of a Chinese speaker's mind does not understand Chinese (and is therefore not conscious) any more than Searle does and cannot constitute a mind. This is a denial of functionalism because a functionalist could create a functional isomorphism between the simulation of a Chinese brain and the brain itself, proving that the former understands all that the latter does including Chinese. Searle believes that the formal rules for manipulating Chinese symbols and the symbols themselves (syntax) do not convey any meaning or understanding (semantics).

Some have criticized the argument on the grounds that Searle has not proven that semantics can't come from syntax, and so his argument already assumes that functionalism is false (Russel 959). But the argument presents a strong challenge to functionalism: how is it that formal rules alone could lead to genuine meaning and understanding? If we can answer this question, we will gain a greater

understanding of functionalism and have an intuitive refutation of the Chinese room argument. Daniel Dennet, among others, has argued that the Chinese room has *derived* meaning because the Chinese symbols that Searle manipulates really do mean something to people outside the room (Cole 5.2). This seems difficult to accept because the room's consciousness should not depend on what is outside the room; a person does not become any less conscious when their native language becomes extinct⁵.

The Nature of Understanding

Can functionalism be strengthened against the Chinese room argument? The alternative to functionalism is to concede that the underlying details of a system *do* matter: that somehow, a brain made of neurons is fundamentally different from one implemented in software⁶. Functionalist ideas such as multiple realizations seem intuitive. It seems that the main problem with functionalism is its vagueness: it is not clear what parts of the human mind are necessary for it to be a mind and which are just human implementation details that could be replaced without affecting mental functioning. Functionalists need a theory of the mind that allows the Chinese room's processes to have intrinsic meaning as a conscious, understanding system. I think functionalism can resist the Chinese room argument if this is clarified.

To start, we haven't determined what intelligence really is. Clarifying this would create a model for an ideal intelligent system that we could use to show that real systems are intelligent based on an isomorphism to an ideal intelligent system. A reasonable intuition is that an intelligent being must *understand*. Searle implicitly assumes this in his Chinese room argument, in which he argues that

⁵ Assuming that the person's experience was the same either way.

⁶ This position is advocated by Searle and is called biological naturalism (Russel 954). It is a difficult conclusion to accept for reasons given in Russel p. 956-958. Essentially, the argument is based on the brain replacement thought experiment I gave earlier: if you gradually replace the biological neurons in the brain with artificial neurons and then back again, the brain must be conscious before and after the operation, and any changes in consciousness during the operation cannot be observed in behavior, so if consciousness fades then it must do so gradually with no observable consequences.

because a machine cannot *understand* what it is doing any more than Searle can understand Chinese, it cannot be a mind. Intuitively, to understand something is to know the meaning behind it. It seems that, whether or not understanding is *sufficient* for intelligence, it is *necessary* and is a large part of intelligence.

This leaves us with another question: what is understanding? We have an intuitive idea of what this is. Understanding involves recognition of the *structure* and *relation to context* of something. For example, to understand this essay, you must understand that it is in English, which involves a recognition of this essay's grammatical structure and the contextual observation that this structure corresponds to English. You must also understand that it is expressing certain ideas, which have some structure in themselves but must also be understood in the context of outside information, such as the knowledge that the brain is made of neurons, or concepts, such as your intuitive grasp of what understanding is. How is it that humans understand, and how might it be possible for computers to as well?

Understanding through Induction

Empiricist philosopher David Hume analyzed human understanding in his book *An Inquiry Concerning Human Understanding*. He shows that we do not have a purely logical basis for understanding the world, noting that “That the sun will not rise tomorrow is no less intelligible a proposition, and implies no more contradiction than the affirmation, that it will rise. We should in vain, therefore, attempt to demonstrate its falsehood” (Hume IV.21). Instead, Hume believes that we can understand processes such as the sun rising through experience. Hume says that “there appear to be only three principles of connexion among ideas, namely, Resemblance, Contiguity in time or place, and Cause or Effect” (Hume III.19). He describes how a person might infer these connections from

experience by, for example, noting that one thing always precedes another (Hume V.35).

Only experience allows us to infer that one thing causes another, and we can use this connection to predict that this relationship will probably hold in the future. This process underlies the scientific method: scientists perform experiments to establish causal relationships between ideas (for example, by observing that water at a higher temperature has higher volume) and form theories to find deeper causes for these relationships (in this case, knowledge about the properties of liquids). Inferring these relationships means understanding our observations through both knowledge about the structure of the observations (noting the correlation between temperature and volume) and context (prior knowledge about liquids). Though Hume does not use the word “induction”, it is used to refer to this process of inferring relationships such as cause and effect. In general, induction deals with inferring general rules from specific instances (for example, inferring that gravity pulls all objects down from seeing it act on a large number of objects). These general rules are a form of understanding because they explain the structure of observations and how they relate to contextual information. In this way, induction can be seen as a form of intelligence based only on sensory impressions.

When scientists look for inductive relationships (such as scientific theories), they prefer simple theories that make few assumptions to complex ones if both are consistent with observed data. This principle is called Occam's razor. When using Occam's Razor, it helps to think of the *model* as being separate from its *instantiation*. The model contains general information that can be used to explain a variety of different situations, while an instantiation of the model applies it to a specific situation. In science, if we want to explain the outcome of an experiment, the model consists of the hypothesized relationship between the independent and dependent variables and the instantiation consists of assumptions that measurement error happened to turn out this way, a truly random quantum process just happened to produce this result, etc. If we want to explain English text, the model is written English (along with general knowledge about what people write about) and its instantiation consists of

the specific ideas expressed and stylistic choices (including mistakes). The model represents our knowledge about the text's structure (English), while the instantiation represents whatever this particular English text happens to be like. Together, knowledge of English text and this specific instantiation of it provide a full explanation of the English text and could be used to perfectly reconstruct it. Alternatively, we could say that the model is random text and the instantiation is exactly the text we are explaining. This is like saying that the text was just random and “happened” to be grammatically correct English. While this model is much simpler than English, its instantiation is much more complex. Occam's Razor should tell us to select the English model instead of the random one. On the other hand, if we saw actual random text, it is better to assume it is random and just happened to turn out that way than to apply the English model and assume that the writer just “happened” to make thousands of typos. We want the combination of the model and its instantiation to be simple. Humans can do this to some extent, but we would like a way to mathematically define what simplicity is so that we could see if computers could use Occam's razor as well.

Algorithmic Information Theory

Algorithmic information theory is a branch of computer science that deals with ways to describe information in terms of programs. Ray Solomonoff founded algorithmic information theory with his formulation of algorithmic probability. Recall that, according to Occam's Razor, simpler explanations are more likely to be continued than complex ones. We need a way to measure complexity, so we might try writing out the explanation in English and comparing the lengths. For example, we might be given the first 8 bits of a binary sequence, “11111111”. If we wanted to explain the 8 bits in a way that lets us predict the rest of the sequence, we would try explaining the 8 ones. We see that “they are all ones” is shorter than “the first 8 are ones and the rest are zeros”, counted by number of characters in the

description, so we would assign the first one a higher probability. Finding a short description of data (*compressing* it) can be seen as a way to explain or understand it. Because it is imprecise, English is not a good language for unambiguously describing strings, so in algorithmic information theory the description takes the form of a program (written in binary in some programming language, such as Python⁷) along with an input string. When it is given this input string, the program should produce the string we are trying to describe⁸. I will call this combination of the program and its input an algorithmic description. Because we can write both the program and its input in binary, we can write an algorithmic description in binary as well by smashing the two together⁹.

Algorithmic descriptions formalize the notion of an explanation. An algorithmic description can be seen as a model along with its instantiation. We might want to model a sequence of random English words (such as “thin earplug shoemakers”), so we give each English word its own binary code¹⁰ (e.g. “100101101111” for “thin”) and write a program that will take the binary codes as input and print out the corresponding English words. If there are enough words given, this approach will result in a shorter algorithmic description than how long the word sequence would be if translated directly into binary. This algorithmic description closely matches the way humans model a sequence of random English words. We model the data as a sequence of English words (represented by the program that prints out English words), but there is no way to say why each word is what it is; it just happens that the first word in the sequence was “thin”. This is instance information, so it is given to the program as input. Similarly, we could think of how to model an image of a circle¹¹. A person would describe it as

7 How do we decide which programming language? It doesn't especially matter, as long as the language is Turing-complete; Solomonoff's convergence theorem implies that ALP using any two languages will arrive at almost the same answer when given enough data (Solomonoff 11).

8 Technically, the program should only be able to read the input one bit at a time and halt on its own accord when it has read the entire input and no more.

9 This requires knowing when the program ends and the input begins, which is equivalent to saying that the program is self-delimiting. I will assume that programs are self-delimiting. Python programs are not normally self-delimiting but they can be if there is a special character that marks the end of the program.

10 This technique of assigning binary codes is called Huffman coding.

11 Even though they don't appear to be linear as text is, images can still be represented in binary.

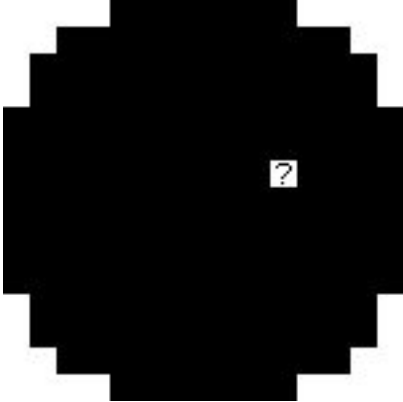
a circle of a certain color, radius, and position. We could algorithmically describe the circle as a program that draws a circle with a certain color, radius, and position, and give these parameters in as input. This description will be very compact: it should not take many bits to write a program that draws circles, and the parameters can be described using only a few numbers each.

Why do humans explain data by finding simple descriptions? One reason is memory: more compact descriptions are easier to remember. It is easier for us to remember the idea of a black circle than to remember the exact value of every pixel in the image, and either description would allow us to recreate the circle. But there must be a more important reason for description. In Jorge Luis Borges' short story "Funes the Memorius", Ireneo Funes is a boy who has a perfect recollection of every sensation he experiences and thus loses the ability to understand abstract ideas:

He was, let us not forget, almost incapable of general, platonic ideas. It was not only difficult for him to understand that the generic term dog embraced so many unlike specimens of differing sizes and different forms; he was disturbed by the fact that a dog at three-fourteen (seen in profile) should have the same name as the dog at three-fifteen (seen from the front). His own face in the mirror, his own hands, surprised him on every occasion.

Because he has perfect memory, Funes has no motivation to come up with abstract, compact descriptions. He simply memorizes every detail of his experiences instead. His memory not only prevents him from understanding abstract language, such as the word "dog": it causes him to lose the ability to anticipate the future. Because he has no general idea of his own appearance that unites every experience he has of seeing himself, even his own face in the mirror surprises him; he cannot anticipate what he will look like the next day. We can see that, in addition to memory, the utility of compact descriptions is in prediction.

If we were given an image of a circle with one pixel missing, we could easily determine the

	<pre> For every integer 0 <= x < WIDTH: For every integer 0 <= y < HEIGHT: If $(x - \text{CENTER_X})^2 + (y - \text{CENTER_Y})^2 < \text{RADIUS}^2$: Color pixel (x, y) INTERIOR_COLOR Otherwise: Color pixel (x, y) EXTERIOR_COLOR </pre>	<pre> WIDTH = 15 HEIGHT = 15 CENTER_X = 7 CENTER_Y = 7 RADIUS = 8 INTERIOR_COLOR = black EXTERIOR_COLOR = white </pre>
	<pre> For every integer 0 <= x < WIDTH: For every integer 0 <= y < HEIGHT: Read the next color in the list. Color the pixel at (x, y) that color. </pre>	<pre> WIDTH = 15 HEIGHT = 15 white, white, white, white, black, black, ..., yellow (for missing pixel), ... </pre>

A circle with a missing pixel and two consistent algorithmic descriptions of it. Model (program) is on the left and instance information (input) is on the right. The first description (top) is closer to how people would think of the circle: the pixels close enough to the center are colored black and the pixels outside this radius are colored white. The second is one of Funes's descriptions: the color of every pixel in the image is specified, and the missing pixel is considered to be yellow (though it could be any color). The top description is given much more weight because it is much shorter, so its prediction (that the missing pixel is black) is more likely to be correct. In reality both the program and the input will be in binary.

missing pixel's color. Although it is possible that the original image had a white pixel in the middle of an otherwise black circle, this is much less likely than the entire circle being black. We should consider ways to describe the *entire image* that are consistent with all the pixels we can see. If we described the image as a black circle on a white background, this description would be consistent with what we see in the image and would tell us that the missing pixel is black. Funes could describe the entire image by saying that the first pixel is white, the second is white, etc. with some arbitrary color for the missing pixel. One of Funes's descriptions (the one that says the entire circle is black, one pixel at a time) is likely to be correct, but the rest (which say that the entire circle is black except for the missing pixel, which is some other color, also one pixel at a time) will probably be incorrect. Funes would not recognize the circle as a whole and would have no reason to think that the missing pixel is black instead of another color. The compact description is much more likely to give accurate predictions than the long descriptions. Perhaps, as Borges suggests, our limited memory is what motivates us to find

compact descriptions of our experiences with the additional effect of helping us anticipate the unknown.

Solomonoff's algorithmic probability (ALP) gives us a way of turning compact descriptions into predictions¹². The algorithmic probability of a piece of data is determined by all of the algorithmic descriptions that describe it, with short descriptions contributing much more to the probability than long programs¹³ (Solomonoff 2). That means that, if we had to predict the color of a pixel in an image of a circle, we would look for descriptions that describe the entire image (including the missing pixel) that are also consistent with the pixels that we can see. These would include but not be limited to both the simple description of the entire image as a circle and Funes's extremely reductionist descriptions. Because the simple description is much shorter, ALP will think that it is quite likely to be correct when applied to the missing pixel¹⁴. Funes's descriptions would be given very low probabilities and so they would not influence the probability a great deal. Thus, ALP would predict that the missing pixel is almost certainly black. Solomonoff proves that “if there is any describable regularity in a body of data, ALP will discover it using a relatively small sample of the data” which makes ALP “the only induction technique known to be complete”. “Describable regularity” includes all patterns that an algorithm could create, such as a circle. It includes any regularity that could be observed in the universe¹⁵. Solomonoff also shows that ALP is incomputable, meaning that it cannot be calculated by a computer with great accuracy; it is always possible that there is a short description of the data that has not been found (Solomonoff 3). ALP can be approximated by RBP (resource bounded probability), which is an algorithm that searches and tests short descriptions (Solomonoff 3). A computer using an

12 This probability is Bayesian, meaning that it represents how likely a program believes something is, which is the probability it should use to make decisions.

13 Specifically, $\sum 2^{-x}$ for every description where x is the length of the description. This means that longer descriptions are given exponentially less weight.

14 This idea has roots in empiricism. For example, William Clifford writes that “we may add to our experience on the assumption of a uniformity in nature”; that is, we can assume that there are rules that underly both the known and the unknown, so the best explanation for what we do know is likely to be correct for what we don't know (Clifford 8).

15 Assuming the strong Church-Turing thesis is true.

approximation of ALP such as RBP can find successively better explanations for the data but can never be sure that it has found the best explanation, much like scientists who can never be certain that they have found the best explanation for the outcomes of experiments but try to find successively better ones. It might turn out that apparently random quantum events follow a deterministic pattern, but it is unreasonable to expect a human or computer to be able to detect it even if it exists¹⁶.

Data compression is the process of finding short algorithmic descriptions of some data. To see how compressing data can be seen as equivalent to understanding it, consider a computer with no knowledge of English that analyzes the text of part of a newspaper article using a method such as RBP. Initially it will not see any pattern in the text at all; it will not be able to find a description for the article shorter than the article itself. It might notice that some letters are more common than others, enabling it to use a technique such as Huffman coding to find a shorter description of the text. It might then notice that text can be split into words and that some words are more common than others. This allows it to find a shorter description by encoding at the level of words rather than letters. With more advanced processing, it could discover the grammar of English and note that the article is written in grammatically correct English. It could use this knowledge of grammar to anticipate the parts of speech of words and further compress the text. By understanding higher-level properties, such as the fact that nearby sentences tend to contain identical or related words, the computer could compress the text even more effectively. If it had many newspaper articles, it could detect general patterns, such as noting that news articles often state sequences of events. In the extreme, it could find deeper patterns behind these sequences of events that explain why they occur. The same reasoning could apply to non-textual data such as images, audio, or video. We can see that the more patterns in the data the computer discovers (or the more it *understands* the data), the more it can compress the data. In short, data

¹⁶ If you could always detect all patterns in random-looking data, you could break public-key cryptography because encrypted data can be described compactly. We can't expect any human or computer to break public-key cryptography just by looking at encrypted data.

compression is a model for understanding: a good data compression system understands the information it is given.

This is a very powerful idea. It is so powerful that a theory of the mind can be based on it. Other writers have advanced parts of this theory of the mind, but I don't think it has been named yet as a whole. So, in the tradition of the theory of the mind, I will give it a name: compressionism. Compressionism is a form of functionalism that includes three central ideas, the first of which is that data compression is equivalent to understanding. I will present the other ideas later, which mostly follow from this one. According to compressionism, any system that compresses information well understands it. We have arrived at the first principle of compressionism by perfecting Occam's razor, the key function of human understanding. As it is a form of functionalism, compressionism directly contradicts the conclusion of the Chinese room thought experiment that no syntactic (rule-based) system can have any understanding. Later I will explain what compressionism tells us about the Chinese room.

Prediction

One might wonder how a data compression system can demonstrate intelligence. This would be required in many cases for compressionism to at least somewhat agree with behaviorism. In the previous example, a computer compressing part of a newspaper article would be able to anticipate what the remainder of the article would look like. By “anticipate” I mean that it has a probability distribution for the remainder of the article: it can tell, given a hypothetical continuation of the article, approximately how likely it is. This does not mean perfectly predicting what the rest of the article looks like; it is more akin to recognizing that a coin is unbiased than perfectly predicting whether it will come up heads or tails. If the compressor knows nothing about the structure of the text, then it will

anticipate that the remainder of the article would be random text, as it detects no pattern. If it knows the different frequencies of letters, it will anticipate that the rest of the article will follow a similar distribution of letters, with more e's and t's than q's and z's. After recognizing words, it will guess that the article would consist of random English words, and after recognizing English grammar, it will guess that the article would consist of random grammatical English sentences. If it detects deeper patterns that explain events covered in the news, it will guess that these patterns will hold true in future news articles.

We can see that the better of a description of the previous text the computer can get, the closer its anticipation of the remaining text will be to reality. This works the same way that better scientific theories allow scientists to better predict the outcomes of new experiments. It is just like predicting the missing pixel in an image: the known pixels correspond to the past and the missing pixel corresponds to the future. The future can be predicted to follow the same rules that the past does just as the missing portion of the image can be predicted to follow the same rules that the rest of the image does. Anticipating the future is a key function of human intelligence and can also be done by a computer using approximations of ALP. According to compressionism, because the computer approximating ALP realizes the functions of explaining data and anticipating the future, it really does understand the data and anticipate the future and would be intelligent in these areas.

What if we could perfectly compute ALP? This is proven to be impossible, but is still interesting to consider because it tells us about approximations of ALP as well. Because ALP lets us know what an entire piece of data might look like if we are given a part of the data, we can take *samples* of what an ALP predictor would think could plausibly fill the missing parts¹⁷. This is akin to asking a person to imagine what comes next in a sequence or what a missing part of an image would look like. If we took samples this way, we would get a good idea of what the predictor thinks will

¹⁷ We can do this by using the continuous version of ALP and randomly generating the sample one bit at a time. This also works for approximations such as RBP.

follow the prefix.

If we gave our ALP predictor the entire history of the Dow Jones Industrial Average (DJIA), it would anticipate its future to the best extent possible by looking at the history alone. It could give the probability that the DJIA would go up or down. We could also have it sample a future for the DJIA as a graph. Of course, this prediction will be limited because the only information the predictor has available is the history of the DJIA. It would be much more accurate if we gave it financial news as well as the DJIA for every day in history. Although the news might be in English, which the predictor does not originally understand, it will learn English (to the extent that is necessary to predict the DJIA) if it is given enough material as with the newspaper article compressor. With the financial news, the predictor would be more accurate than any investor who lacks access to insider information.

Predicting quantities such as the DJIA would be very useful in practice, but the capabilities of an ALP predictor extend into areas such as creativity, imagination, and imitation. We could give the predictor every patent issued by the US Patent and Trademark Office and ask it to give us samples of future patents. These samples represent patents that the predictor thinks the USPTO will issue in the future. They will be based on whatever patterns the predictor discovers in the patents. The predictor should be able to learn ideas such as English, science, and knowledge of what is useful and patentable by looking at millions of patents; these ideas certainly help to explain them. If this turns out to be insufficient, we can always give the predictor more data, the kind that patent examiners would already know when they evaluate patents. Based on these ideas that explain existing patents, the predictor will create a new patent that follows these patterns. I think that most of them would satisfy the requirements for patents well enough to be issued; if any are not issued, this would reflect the USPTO's inconsistency because the patents created by the predictor would demonstrate the same patterns present in previous patents. An ALP predictor could create many patentable inventions, an extremely useful task that shows creativity.

Similarly, we could have the predictor generate popular music by looking at popular songs (in a binary format such as WAV) and generating new songs to continue the patterns found in popular music. Continuing patterns may look like imitation of a style, as the style represents the patterns we think about when describing music. But the predictor will find deeper patterns, such as synthesis and evolution, which also underly popular music. Sampling using algorithmic probability can be seen as a form of imagination because the predictor can imagine what music will plausibly be like in the future. Imagination can also deal with hypothetical situations: we can give the predictor a fabricated history, such as a historical record in which an important detail is different, and ask it to imagine how events would continue from that point. We could even give it some conceptual drawings of a fictional world and have it create a science fiction movie out of them¹⁸.

It turns out that an ALP predictor can pass the Turing test (Mahoney 3-4). We can give it the records of all previous contest transcripts between judges and humans. From these it will infer all patterns necessary to explain these conversations, which requires extremely accurate knowledge of human psychology. Then, we can ask it to give a sample answer to the judge's question based on the conversation so far. For example, if we gave it the conversation “Hello. How are you? / Good. / What color is the sky? /”, it would give the kind of answer that would follow this, which is probably something like “Blue.”. This response will follow almost¹⁹ the same distribution that a response from a human would, so a predictor having a conversation this way would be indistinguishable from a human and pass the Turing test.

Of course, we cannot compute ALP perfectly, so all of this is fantasy. However, as approximations get successively closer to ALP, they will approach this perfect prediction, creativity,

18 This requires knowing the relationship between concept art and the movies based on them. We could give the system some examples of science fiction movies and their concept art.

19 The predictor's distribution approaches the true distribution as it is given more conversations. See Solomonoff's convergence theorem (Solomonoff 11).

and imitation²⁰. A program that recognizes many, but not all, patterns in human textual conversations and imitates them might be very difficult to tell apart from a real human. A program that recognizes some patterns in music and continues them might be able to imitate a particular style without recognizing or replicating subtler patterns in music such as evolution. The main obstacle to very good approximations of ALP is computation time: algorithms such as RBP take a very long time to find short algorithmic descriptions. Less general methods tend to be much more efficient even though they fail to recognize many patterns and are often used in machine learning. We can see that good approximations of ALP (data compression systems) exhibit many of the abilities that intelligent beings have. Behaviorists would agree that good data compression systems really do have creativity and imagination as they can demonstrate these abilities. Functionalists can only be convinced if data compression systems implement the functions of understanding, but as previous reasoning shows, functionalists should consider data compression to be *the* function of understanding: we arrive at it by perfecting Occam's razor, the central idea behind human understanding.

Explaining Consciousness

Phil and Rebecca Maguire advance compressionism in their paper “Consciousness is Data Compression”. They assert that “All successful predictive systems, including animals and humans, are approximations of algorithmic induction. All useful contributions to human knowledge work by coaxing people into modifying their inductive strategies in such a way that they better approximate algorithmic induction” (Maguire 749). They justify this idea by arguing that “In order to thrive in an uncertain environment, organisms must be able to anticipate future events; the more efficiently they can

²⁰ We can define how good an approximation of ALP is by how much it compresses the data. The better the compression ratio is, the better it understands the data. This definition has the problem that it might emphasize explaining low-level details over high-level details because they affect the compression ratio more, though high-level details are more important in practice. I don't know if this will have to be accounted for in practice. See “Towards a Universal Theory...” for more discussion of decision-making based on approximations of ALP (Hutter).

compress their experiences, the more accurate these predictions will be” (Maguire 749). In the real world, the data that organisms compress is not usually text but sensory data from the environment including vision, sound, and touch. Having a better compression of this information means having a better understanding of the environment and allows for better predictions. These predictions are essential to survival; they allow the organism to anticipate where food might be located, for example²¹. As algorithmic probability is the only complete induction method, it will recognize all meaningful patterns. Other methods approach algorithmic probability as they recognize more patterns. By recognizing more patterns, better data compression systems enable organisms to model and predict their environments better, creating an evolutionary selection for better data compression.

The Maguires assert what they call the consciousness conjecture, the idea that “the experience people describe as consciousness is equivalent to the compression that the brain carries out” (Maguire 749). Note that this equivalence is symmetric: the consciousness conjecture asserts that consciousness and data compression are indistinguishable, implying that all sufficiently advanced data compression is consciousness and that all consciousness is advanced data compression. The consciousness conjecture seems like a very strong statement. The Maguires argue that the brains must be good at compressing data because this process is required to anticipate the future, an ability that greatly improves an organism's chances to survive and reproduce. The brain must take many types of sensory data as well as memory and compress them in parallel to get a unified model of the environment over time and, according to the consciousness conjecture, consciousness (Maguire 749).

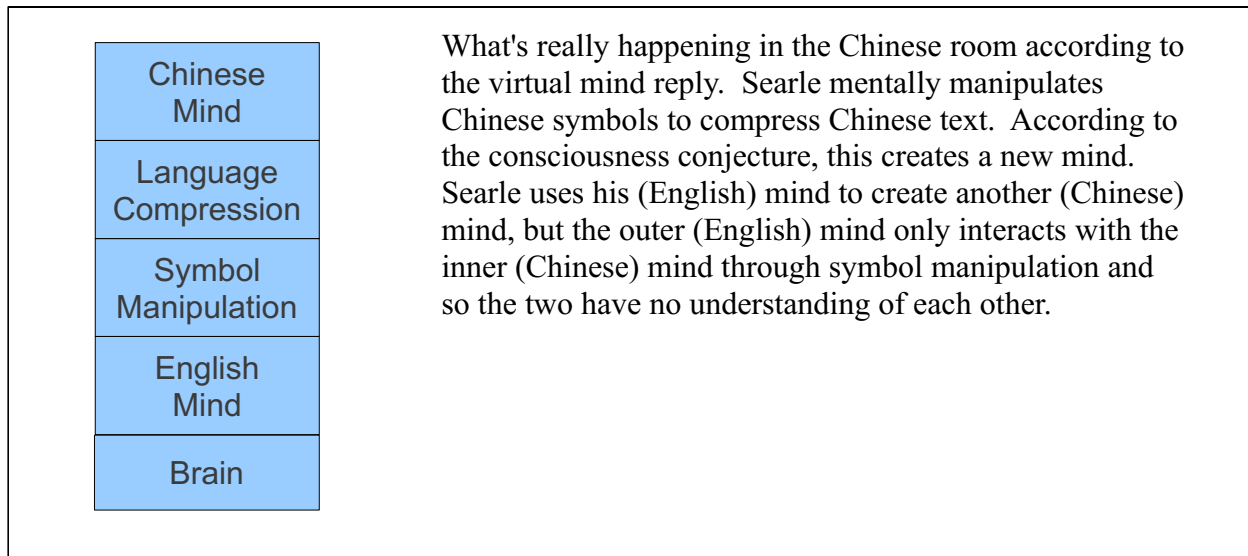
The Maguires show that the consciousness conjecture is consistent with our subjective

21 Technically, ALP only allows predicting where you might see something that looks like food; that is, it only allows predictions of sensory input. However, data compression systems will get a picture of the part of reality that it observes that improves with better data compression, just as a text compressor will get a picture of the processes that went into creating the text. Some form of motivation, such as a survival instinct or altruism, can be based on this idea of reality. Motivation based only on anticipated sensory input would amount to solipsist hedonism, though it can still be useful (see reinforcement learning). So, though I glossed over this detail, I think there is more to ALP-based understanding than prediction of sensory input.

experience of consciousness. If a compression system compresses data from its environment, one might wonder where self-awareness comes from. The Maguires first explain that the brain must have a model for a person in order to explain redundancies in the behavior of others (Maguire 750). This model for a person explains why other people behave the way they do (for example, why they act as if they have memory and are self-interested). The brain can then apply this model to itself, helping it explain its own behavior and creating self-awareness.

The consciousness conjecture also provides a model for qualia (subjective experience, such as the experience of the color red). The Maguires explain that qualia have to do with exactly *how* our experiences are compressed, not the objective experiences themselves (Maguire 750). A person may use a particular mental process to encode the objective experience of red light. This process will be consistent over time, allowing the person to recognize that two objects are both red. However, it is not guaranteed to be consistent between people, and even if the exact process were described in a book, a person could not use the text to replicate the process representing the subjective experience of “red” in his or her mind (Maguire 750). This explains why knowing about color is not the same as experiencing it. In data compression systems, the particular wavelength of light produces red gets assigned a particular code (like binary codes for words in the text compression example) that the system ensures is consistent over time.

The Maguires' consciousness conjecture is closely tied to the first central idea of compressionism, which views data compression as equivalent to *understanding*. If consciousness is to be interpreted as understanding of one's experiences and self, then the consciousness conjecture is a consequence of compressionism, and so I will consider it to be the second central idea of compressionism. With this addition, compressionism is a very promising refinement of functionalism that reduces intelligence and consciousness to a single function that can be realized in many different ways. It appears to be consistent with properties of consciousness such as self-awareness and qualia,



strengthening its credibility.

Semantics from Syntax

This leads us back to the Chinese room argument. How can the compressionism explain how subjective experience and understanding can come from formal rules? We must look at what happens when the Chinese room implements a data compression system. Let's give Searle the transcripts of millions of Chinese conversations. We will also have him record the conversation he has with the outside world as it happens. Whenever he wants to say something, Searle should use an algorithm such as RBP to compress all of the previous transcripts as well as his current conversation. According to the first principle of compressionism, the Chinese room now understands Chinese textual conversations, including the one it is currently having. Searle should then sample from future anticipated conversations to tell him what he should say to continue the conversation. If its approximation of ALP is good enough, Searle's algorithm will produce a response indistinguishable from that of a Chinese speaker.

The consciousness conjecture says that there is a separate consciousness in the Chinese room. This consciousness is a product of the processes Searle uses to manipulate the characters and it

understands Chinese textual conversations, including the one it is having. It has self-awareness because it can explain its own behavior (as it compresses its own conversation). It can be said to understand Chinese as much as a person whose only interaction with the world is text can understand Chinese. Searle has created a separate consciousness that has very little to do with his own, as its compression is completely separate from his. Even if he memorizes the rules and symbols, Searle has created a mind that shares his brain but has no interaction with his ordinary mind (which understands only English)²². Searle anticipates an argument like this (which he calls the “virtual mind reply”) and responds:

According to one version of this view, while the man in the internalized systems example doesn't understand Chinese in the sense that a native Chinese speaker does (because, for example, he doesn't know that the story refers to restaurants and hamburgers, etc.), still "the man as a formal symbol manipulation system" really does understand Chinese. The subsystem of the man that is the formal symbol manipulation system for Chinese should not be confused with the subsystem for English. ... The subsystem that understands English (assuming we allow ourselves to talk in this jargon of "subsystems" for a moment) knows that the stories are about restaurants and eating hamburgers, he knows that he is being asked questions about restaurants and that he is answering questions as best he can by making various inferences from the content of the story, and so on. But the Chinese system knows none of this. Whereas the English subsystem knows that "hamburgers" refers to hamburgers, the Chinese subsystem knows only that "squiggle squiggle" is followed by "squoggle squoggle." All he knows is that various formal symbols are being introduced at one end and manipulated according to

²² Of course, memorizing and operating on millions of transcripts is beyond any human's ability, so the only way Searle could do this is if he had an extremely large brain. It is more plausible that such a large brain could house more than one mind.

rules written in English, and other symbols are going out at the other end.

When Searle refers to the “English subsystem”, he is referring to a previous example he introduced in which Searle answers English questions about stories using his normal mental abilities. We can see that the English subsystem really does understand the stories he is given: he knows what the word “hamburgers” means. Searle argues that the Chinese subsystem (the part of his brain that does symbol manipulation) does not have this same understanding of the Chinese text. But Searle's language reveals that he fundamentally misunderstands the virtual mind reply: he presents the Chinese subsystem as *a part of his own mind* instead of a completely different mind. He uses the word “he” to refer to it and asserts that it only understands Chinese as squiggles and squoggles. These assumptions would be true if the virtual mind were a part of Searle's conscious mind, but the virtual mind reply should really say that the virtual mind is a completely different mind that emerges from the symbol manipulation, just as it could emerge from neuron firings. It doesn't make sense to say that a bunch of neurons understands anything any more than it makes sense to say that Searle's symbol manipulation system understands Chinese. We should study the system implemented by these lower-level processes, not the lower-level processes themselves.

If we do this, what kind of picture do we get of the Chinese subsystem? The Chinese subsystem understands many statistical patterns in Chinese textual conversations. It will know that certain words are more likely to follow previous words just as English speakers know that “The sky is” is likely to be followed by “blue”. Of course, “blue” means more to most English speakers than its relations to other words and groups of words. It is associated with the sensory experience produced by a certain wavelength of light. A simple adaptation of the Chinese room gives the Chinese subsystem visual and auditory data along with the conversations it knows about; this would allow it to build up these associations and fully understand Chinese²³. The Chinese subsystem knows all important patterns in

²³ See Searle's discussion of the robot reply (7-8). I actually believe that a mind with no sensory experience outside of text can still understand text, just as a deaf person can understand text about sound and physicists can understand text about

Chinese text, which, as I have argued earlier, is equivalent to understanding it; according to the consciousness conjecture, it would also be conscious. Searle might reply that, when I say that the Chinese subsystem performs data compression and understands patterns, this is only my interpretation of the Chinese subsystem and has no objective basis.

Objective Meaning

Occam's razor (and by extension, ALP) tells us that some explanations (or interpretations) are better than others. It is an objective standard that distinguishes reasonable interpretations from arbitrary ones. Searle believes that there is no such objective standard in the case of computer systems (Searle, "Is the Mind a Digital Computer" 35-36):

Syntax, in short, is not intrinsic to physics. This has the consequence that computation is not discovered in the physics, it is assigned to it. Certain physical phenomena are assigned or used or programmed or interpreted syntactically. Syntax and symbols are observer relative. It follows that you could not discover that the brain or anything else was intrinsically a digital computer, although you could assign a computational interpretation to it as you could to anything else.

Searle tells us that nothing is intrinsically a digital computer: the digital computer is just a metaphor that we subjectively assign to certain systems. In the extreme, *any* sufficiently large physical system can be interpreted as *any* computational system; the wall behind Searle's back can be interpreted as a computer running the Wordstar program (Searle 27). This idea challenges functionalism: how can we reasonably talk about functional isomorphisms when a human brain can be seen as isomorphic to a

black holes. Understanding text in a language can be considered different from understanding the language itself, but there is understanding nonetheless. We can imagine a man who initially knows no language, is confined to a room, and communicates with Chinese speakers through text messages. If he is intelligent enough, he will learn Chinese well enough to have conversations; he will learn all important associations between words. This is the kind of understanding that the Chinese subsystem in the original (non-robotic) Chinese room could have.

rock²⁴? Searle would doubt my argument that the Chinese subsystem really is a data compression system, because the idea of a data compression system is just a metaphor that I find useful to explain the Chinese subsystem.

This argument flies in the face of intuition. We know that some interpretations are better than others: it makes a lot of sense to consider a physical computer running Wordstar to actually be a computer running Wordstar, and it makes absolutely no sense to say the same about a wall. It seems that a statement like “this computer is running Wordstar” is as objectively true a statement as “this image is approximately a circle” or “this tofu is made of atoms”²⁵: though all these statements are interpretations, they are all obviously true and I think there is an objective basis for saying this.

These sentences are abstract, partial explanations for the physical objects. We can imagine the explanation being extended to explain all details of the objects, down to the positions of every subatomic particle. A description of a physical computer running Wordstar can first describe the high-level details (that it is a computer running Wordstar) and then describe exactly how these details are realized²⁶. The length of this description will be very close to the length of the best description possible for the entire system because the two are independent: the program's running itself has nothing to do with how the computer happens to store memory. On the other hand, if you had to describe a wall as a computer running Wordstar, you would have to specify exactly the relationship between the positions of molecules in the wall and states of the Wordstar program. This description would end up being significantly longer than an ideal description of the wall. We can determine how good a high-level description of something is by how much a description based on it differs from an ideal description.

24 Putnam himself eventually rejected functionalism and gave this argument (Chalmers 2). Chalmers responds by restricting the notion of a functional isomorphism but admits that his system excludes certain isomorphisms that should be valid such as virtual memory (Chalmers 27).

25 Atoms don't “really” exist in physics: they're just groups of subatomic particles arranged in a certain way, so the idea of an atom is only an interpretation of the group of subatomic particles.

26 Compare this to a text compressor: the compressor could describe English text as a sequence of words and separately describe what letters are in each word. The logic of determining what word follows other words is mostly separate from the logic of determining what letters form each word.

This reasoning confirms the intuition that it makes more sense to describe a physical computer as a digital computer than it does to describe a wall as a digital computer. This point is so important that I will name it as the third central idea of compressionism: a high-level description is valid if it is consistent with a near-optimal description of the data²⁷.

As a special case, we can look at cases where a high-level description of a physical system as a data compression system (which takes sensory data and finds short descriptions of it) makes sense. According to the consciousness conjecture, this would determine which systems are conscious and which are not. Of course, no one has a god's-eye view of physics or a perfect way to compute ALP, so the third principle of compressionism is only a guide to tell when high-level descriptions are valid, not a foolproof method. Still, we can guess that, for example, systems that behave or look like data compression systems can be described well this way. Compressionism gives us a way to determine when abstractions such as objects or ideas objectively exist or are just subjective interpretations.

Conclusion

Compressionism develops greatly on functionalism. Functionalism tells us that understanding and consciousness can be created in multiple ways, but the compressionism specifies exactly what intelligence and consciousness are by giving an ideal description of them as data compression systems. Compressionism also tells us how to evaluate high-level descriptions so we can compare real systems to data compression systems. These central ideas of compressionism provide a comprehensive response to Searle's Chinese room argument by specifying how semantics can come from syntax and physics.

Compressionism considers all legitimate methods humans use to understand the world

²⁷ I haven't specified this exactly; it's still an open question how much the descriptions are allowed to differ. There might be other, superior methods to define what a good high-level description is, but they will all probably use a form of data compression as this is the best tool we have to theorize about descriptions.

(including natural science, social science, art, and math) to be forms of understanding because they aid in data compression²⁸. It allows computers to be conscious if they are good enough at compressing data, though this is not an easy task given how difficult good data compression is. Compressionism reaffirms the intuition that some explanations are objectively better than others and that high-level concepts such as computers and societies really do exist instead of being mere subjective interpretations. In practice, compressionism suggests that we should continue to develop machine learning algorithms to allow computers to understand data and use this understanding to predict new data. Perhaps we can help computers understand data by having humans translate their understanding of parts of the world into a form that computers can use, similar to an algorithmic description. This would allow computers to explain new data using human-developed models and perhaps build off these models rather than reinventing the wheel. This is probably as close to strong AI as we can get with current hardware, though general intelligence will become possible in the future as hardware improves. Compressionism sees intelligent systems as data compression systems in order to better understand and, eventually, replicate them.

²⁸ I have already compared natural sciences to data compression. Social sciences can be seen similarly in that they look for high-level patterns in human and societal behavior. Art can be seen as a tool that allows people to see the world in new ways, finding shorter (better) descriptions of their experiences. Math is trickier because it is often considered deductive rather than inductive. However, math does give people insights into phenomena and helps in prediction, so it can be seen as a form of induction. See “Driven by Compression Progress...” for more explanation of human pursuits as data compression, especially in relation to art (Schmidhuber).

Works Cited

- Borges, Jorge. "Funes, the Memorius" (1942). Trans. Anthony Kerrigan (1952). *Ficciones*. Web. 20 Mar. 2011 <<http://www4.ncsu.edu/~jjsakon/FunestheMemorious.pdf>>.
- Chalmers, David J. "Does a Rock Implement Every Finite-State Automaton?" (1996). Web. 26 Mar. 2011 <<http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.33.5266>>.
- Clifford, William Kingdon, William James, and A.J. Burger (Ed.). "The Ethics of Belief". Web. 23 Mar. 2011 <http://people.brandeis.edu/~rind/bentley/Clifford_ethics.pdf>.
- Cole, David. "The Chinese Room Argument", *The Stanford Encyclopedia of Philosophy (Winter 2009 Edition)*, Edward N. Zalta (ed.). <<http://plato.stanford.edu/archives/win2009/entries/chinese-room/>>.
- Hume, David. *Enquiries Concerning the Human Understanding and Concerning the Principles of Morals*, ed. L. A. Selby-Bigge, M.A. 2nd ed. Oxford: Clarendon Press, 1902. Web. 20 Mar. 2011 <http://oll.libertyfund.org/index.php?option=com_staticxt&staticfile=show.php%3Ftitle=341&Itemid=28>.
- Maguire, Phil and Rebecca. "Consciousness is Data Compression" (2010). Proceedings of the Thirty-Second Conference of the Cognitive Science Society, Portland, Oregon. Mahwah, NJ:

Lawrence Erlbaum Associates. Web. 20 Mar. 2011

<<http://palm.mindmodeling.org/cogsci2010/papers/0270/paper0270.pdf>>.

Mahoney, Matthew V. "Text Compression as a Test for Artificial Intelligence" (1999). Web. 20 Mar. 2011

<<http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=48FD95AEB58E307FC2B4202A7F1B1AD0?doi=10.1.1.68.2996&rep=rep1&type=pdf>>

Putnam, Hilary. "Philosophy and our Mental Life" (1975). *Mind, Language, and Reality*. Cambridge University Press.

Russel, Stuart and Norvig, Peter. *Artificial Intelligence: a Modern Approach*. 2nd ed. New Jersey: Pearson Education, Inc. 2003.

Searle, John R. "Is the Brain a Digital Computer?" (1990). Presidential Address to the American Philosophical Association. Web. 16 Feb. 2011

<<http://www.ecs.soton.ac.uk/~harnad/Papers/Py104/searle.comp.html>>.

Searle, John R. "Minds, brains, and programs" (1980). *Behavioral and Brain Sciences* 3 (3): 417-457.

Web. 20 Mar. 2011 <<http://citeseer.ist.psu.edu/viewdoc/download?doi=10.1.1.83.5248&rep=rep1&type=pdf>>.

Schmidhuber, Juergen. "Driven by Compression Progress: A Simple Principle Explains Essential Aspects of Subjective Beauty, Novelty, Surprise, Interestingness, Attention, Curiosity,

Creativity, Art, Science, Music, Jokes” (2008). Web. 26 Mar. 2011

<<http://arxiv.org/abs/0812.4360v2>>.

Solomonoff, Ray. “Does Algorithmic Probability Solve the Problem of Induction?” (1997). Web. 6

Feb. 2011 <<http://world.std.com/~rjs/pubs.html>>.

Turing, Alan M. "Computing Machinery and Intelligence." *Mind* (1950). Web. 20 Mar. 2011

<<http://loebner.net/Prizef/TuringArticle.html>>.

Turing, Alan M. (1936). "On Computable Numbers, with an Application to the

Entscheidungsproblem". *Proceedings of the London Mathematical Society*. 2 42: 230–65.

1936–37. Web. 20 Mar. 2011 <<http://plms.oxfordjournals.org/content/s2-42/1/230.extract>>.