A Definition of Information

CS 480
Computational Theory

Adapted from Mind Tools by Rudy Rucker

- Information is data interpreted (or communicated).
- Your hand is information!

  How so? It's designed according to certain instructions coded up in your DNA. The length of these instructions gives a measure of the amount of information in your hand.
- Information is pattern, structure, form.

Some patterns are more complex, hence more interesting than others. Five minutes of silence over the phone tells you a lot less than thirty seconds of animated conversation. If a blank page seems uninteresting, it is because it carries little complexity.

Descriptive Complexity

Define the *descriptive complexity* of the information content of some object to be the *length* of the shortest list of instructions for building a copy of the object.

Under this definition, a random-seeming mess would have a lot of “information,” as it would be hard to give a short description of it. An unmown, weedy lawn would have more information than a golf green.

Information content is, in a sense, related to the degree of unpredictability of the message it presents.

For Example

A novel by Vladimir Nabokov carries more information than does a Gothic romance; that is, if a single word is missing from a Nabokov book, it might take a dozen guesses to ascertain the word, whereas a word missing from a Gothic romance can probably be guessed in two tries.

Suppose you find a string of zeros and ones. How much information is the string giving you? (This is a tricky question.)

A random string of N zeros and ones many contain N bits of information but a string of 32 ones —
11111111111111111111111111111111 —

contains no more information than the number 32, which only takes log 32 (five) bits to write out.

Pretty Random

If the string actually codes up an important message, then it will probably look pretty random. Here's one code:

Space	00000	h	01000	p	10000	x	11000
a	00001	i	01001	q	10001	y	11001
b	00010	j	01010	r	10010	z	11010
c	00011	k	01011	s	10011	Capitalize	11011
d	00100	l	01100	t	10100	.	11100
e	00101	m	01101	u	10101	,	11101
f	00110	n	01110	v	10110	’	11110
g	00111	o	01111	w	10111	Return	11111

Given this code, what does the following string say?

110110100100110000001100101111101010000000011000010111000000
100100010100001001000000010100010000100110011111010000011001
011111010111110100100010100000100110110100001100101010011100

If You Can Read This, You're Smart

(Or at least, you have the code.)

But look at the string as if you didn't have the code. It looks quite random. How could people ever communicate information if they had to use such a weird code? But, wait. We **do** use a weird code.

Have you ever looked closely at Arabic or Cambodian writing? It looks like a bunch of crazy squiggles.

The singer Laurie Anderson is said to have a theory that the Japanese don't really have a language; they just make noises and draw little pictures,

which is exactly what *we* do!

Well-Known

Here is the zeros-and-ones code for a very well-known song. To make life a little easier, I'll put spaces between the blocks of five.

11011 01000 00001 10000 10000 11001 00000 00010 01001 10010 10100 01000
00100 00001 11001 00000 10100 01111 00000 11001 01111 10101 11101 11111

11011 01000 00001 10000 10000 11001 00000 00010 01001 10010 10100 01000
00100 00001 11001 00000 10100 01111 00000 11001 01111 10101 11101 11111

11011 01000 00001 10000 10000 11001 00000 00010 01001 10010 10100 01000
00100 00001 11001 11101 00000 00100 00101 00001 10010 00000 10010 00101
00001 00100 00101 10010 11101 11111

11011 01000 00001 10000 10000 11001 00000 00010 01001 10010 10100 01000
00100 00001 11001 00000 10100 01111 00000 11001 01111 10101 11100

If you look closely, you'll see that there's a log of what information theorists call "redundancy" in this message. Parts of it repeat. Not to keep you in suspense, the message is

Happy Birthday

Happy birthday to you,
Happy birthday to you,
Happy birthday, dear reader,
Happy birthday to you.

The letter code takes up less space but is basically no more efficient than the code which consists of 505 zeros and ones. Sending letters costs some five bits per symbol, so one hundred letters is no more efficient than five hundred bits, but there *is* a shorter way to send the message:

Happy birthday to you . . . chorus,
chorus,
H. b., dear reader,
chorus.

Shorter and Shorter

The shorter English version leads to a shorter code, 345 bits. Of course, this shorter version will be understood only if the receiver understands the special use of the word “chorus.” Suppose we make a somewhat stronger assumption about the receiver’s knowledge. Suppose we assume that the receiver realizes that “Happy Birthday” is the name of the familiar song. Then the message can be very compactly English coded as “ ‘Happy Birthday’ for ‘reader.’ ” This has a compact binary code consisting of 160 bits.

If I want to send the message to you still another time, it is enough for me just to say “the song” — this takes forty bits.

By now, if I *still* want to send the message to you, I can just say “it” — that’s ten binary symbols. If all I’m ever going to send you again is the same Happy Birthday message, and you know this, then I could use the most efficient possible code, and send a single bit every time I want to send my message.

How Many Bits?

So how many bits of information does the Happy Birthday song contain? The simplest measure is the number of zeros and ones it takes to send the message the first way, fully written out. This was, recall, 505 bits. We might call this number the message’s “presentation information.”

Just about anyone can reconstruct the message from the second version, which only takes 345 bits. The 345-bit message can be thought of as an “algorithm,” or set of instructions, for generating the full message. The 160-, 40-, 10-, and 1-bit versions are still more compact algorithms that presuppose increasingly large amounts of knowledge on the part of the receiver.

For any given receiver R, a message’s complexity is the smallest number of bits R needs to guess the whole message. [. . .]

Let R be the universal receiver of messages, whose job is to algorithmically decode them. R is a Universal Turing Machine.

Everything is information.

Mind Tools, Conclusion:

My purpose in writing Mind Tools has been to see what follows if one believes that everything is information. I have reached the following (debatable) conclusions.

1. The world can be resolved into digital bits, with each bit made of smaller bits.
2. These bits form a fractal pattern in fact-space.
3. The pattern behaves like a cellular automaton.
4. The pattern is inconceivably large in size and in dimensions.
5. Although the world started very simply, its computation is irreducibly complex.

So what is reality, one more time? An incompressible computation by a fractal cellular automaton of inconceivable dimensions. And where is this huge computation taking place? Everywhere, it's what we're made of.

Information *Is* Complexity

Assuming any "information" can be quantified, then assigned a number, then represented as a sequence of 1s and 0s:

Information content equals the size of the smallest (shortest) description of that content.

- precise and unambiguous
- recreatable from the description alone