An Interlude: Meaning in Binary Strings

About Meaning and Distortion

While we understand a lot about sound and light, we know very little about how they carry meaning. Why does a piece of sound make you happy, or sad? Why do we say a picture is "scary" or "beautiful"?

In computers, all information is stored as strings of zero and one bits. Whether it is a picture, some text, sound, or light, it is all represented as strings. Where and how do strings carry their "meaning"? If we change some of the zeros and ones will the meaning change?

We don't know the answers to these questions, but we have some guesses. Enclosed is a purely experimental piece of work. If you find the contents interesting, read through it. Maybe you will be able to think of some answers that others have missed!

MEANING IN BINARY STRINGS

Abstract

This chapter deals with the existence of meaning in binary strings. A set of experiments to determine where and how meaning resides in such strings is described. It is found that a string with meaning is indistinguishable from a randomly generated one. The paper examines the postulate that the presence or absence of a byte has little bearing on the meaning of a binary string. Further experimentation seems to show that consciousness is a prerequisite for the creation and perception of meaning in binary strings. The paper proposes a timeless matrix where the interactions of strings produce finite and conscious perception of meaning.

The Meaning of "Meaning"

In this chapter we report our findings on a set of experiments performed on binary strings. In the process, we find that we are plunged into a discussion on meaning and consciousness.

If indeed Cognitive Science has reached a stage where it can say something about these subjects, we hope this chapter will be of use in the formulation of an eventual physics of consciousness.

Introduction

Computers process information using binary strings. All information in current digital computers is stored and manipulated in the form of strings of zero and one. All data consists of such strings, as do all programs that manipulate data. The operating systems of digital computers consist of binary strings, as do their memories. There is no evidence that current computers have any awareness of the meaning of the data they process. On the other hand, the human beings that use the results of data processed by computers deal essentially with the meaning that these strings of zero and one contain. We have attempted to determine where and in what form meaning exists in binary strings.

Definition of "Meaning"

We will define meaning as an evocation (Simon, 1995) of links in memory. For instance, when a reader views words in a text, certain symbol structures stored in the reader's memory are activated. Moreover meaning is associated with recognition (Feigenbaum and Simon, 1984), that is, the evoked structures are recognized and in the process understood or associated with the input text.

In this chapter, we will see how the meanings of binary strings are dependent on expression. That is, a string will have meaning only when expressed through an appropriate method.

BINARY STRINGS

Definition

We will define a binary string as a linear sequence of bipolar states. Zero and One (or One and Minus One) are the most common form in which binary strings exist in computers. However, the term itself need not be defined only with respect to computers. Presence or absence of anything could constitute a bipolar state and, as such, strings are widespread throughout real and imaginary spaces.

Computers and Binary Strings

Digital computers are essentially collections of switches and are, therefore, totally dependent on bipolar states for their functioning. Data of any kind, (numbers, text, sound, graphics etc.,) consist of binary strings. Such data is manipulated through sequences of instructions (programs) that are expressed as binary strings. All input and output is through binary strings under the control of an Operating System, which itself is a program and, therefore, a binary string.

All binary strings in a computer are labeled using "headers" that describe the nature, structure, and function of these strings. Headers are also binary strings.

Bit Patterns in Binary Strings

Binary strings in computers are usually combined into bytes (groups of eight bits, each bit

being a zero or a one). There are 256 bytes that constitute the "alphabet" of the digital computer. The assignment of a character or a symbol to a particular group of eight bits is arbitrary and decided by the American Standard Code for Information Interchange (ASCII). However, the complete ASCII set, if represented as one string, would contain the same number of zeros and ones.

Experiment 1—Bit Patterns in Sounds Pictures, and Programs

In this experiment, we counted the number of zeros and ones in strings that represented sounds, graphics, and programs. In order to do so, we used a small program to skip the target files header, read the data bytes, convert these to bits, and total the zero bits and one bits. A part of the results are shown in Table 1.

Observations

As may be expected, the distribution of 0 and 1 bits appears close to 50:50 in most files. It may be noticed that for certain types of files, e.g., "WAV" (sound) and "BMP" (picture files), this distribution is sometime violated. This is not considered significant as the effect can be generally explained as an artifact. For example, the most common byte in a sound file is the silence byte (1111 1111). Hence, any sound file will show a larger proportion of 1 bits. However, since the choice of the ASCII character 255 as the silence byte is arbitrary, its effect on the distribution is merely an artifact. In fact, if the silence byte is removed, the distribution shifts more toward the 50:50 proportion.

On the basis of the data and the above observations, we assume that the proportion of 0 and 1 bits are the same in meaningful and meaningless strings. We state that: A string with meaning is indistinguishable from a randomly generated one.

In this context, it is interesting to note that it is impossible to determine the nature of a string (i.e., sound, picture, or program) from a segment of it, even if the segment is large. In some instances one may be able to guess from artifacts such as the proliferation of the silence bit, but only if one knew from where to begin counting the bytes. If the string segment does not start from the beginning of a byte, there would be no way to tell what the segment might mean.

Table 1 Distribution of Zero and One Bits in Files

File name	Description	Size (bits)	Percent 0	Percent 1
excel.exe	Program	44,796,032	55.69	44.31
winword.exe	Program	42,543,232	56.59	43.41
alarm.exe	Program	89,472	73.88	26.12
Bitbybit.c	C Program	15,120	53.45	46.55
Showfile.c	C Program	35,040	54.10	45.90
arcade.bmp	Bitmap	5,040	52.34	47.66
castle.bmp	Bitmap	6,224	50.13	49.87
setup.bmp	Bitmap	307,696	39.49	60.51
rec45.wav	Wave	1,005,520	48.60	51.40
beep.wav	Wave	46,112	48.88	51.12

Byte Patterns in Binary Strings

With the above observations in mind, we decided to investigate the distribution of the 256 ASCII characters in meaningful strings. We used another program to determine the distribution of bytes in the same files whose bits had been counted earlier.

Experiment 2—Byte Patterns in Sounds, Pictures, and Programs

In this experiment, we took the target files, removed the headers and counted the bytes. For each file, a plot of the frequency of occurrence of bytes against the ASCII numbers of the bytes was produced. Typical plots for sound, graphic, and program files are shown in Figures 1, 2, and 3.

Observations

Sound files have a distinct signature as far as the frequency of occurrence of bytes is concerned. The dominant byte is the one representing silence. This is due to the nature of audio information as it is perceived by the human ear. Without a liberal punctuation with silence, the meaning of a sound is not obvious. This would be so in text as well, if one were to count the blank character as a part of the text. However, if one treats the silence byte and the blank character as artifacts, the distribution of the other bytes follows no discernible pattern. Nor is there any great variation from one audio file to another.

In the case of graphic files and computer program files, no pattern can be seen. We state that: The presence or absence of a byte has little bearing on the meaning of a binary string.
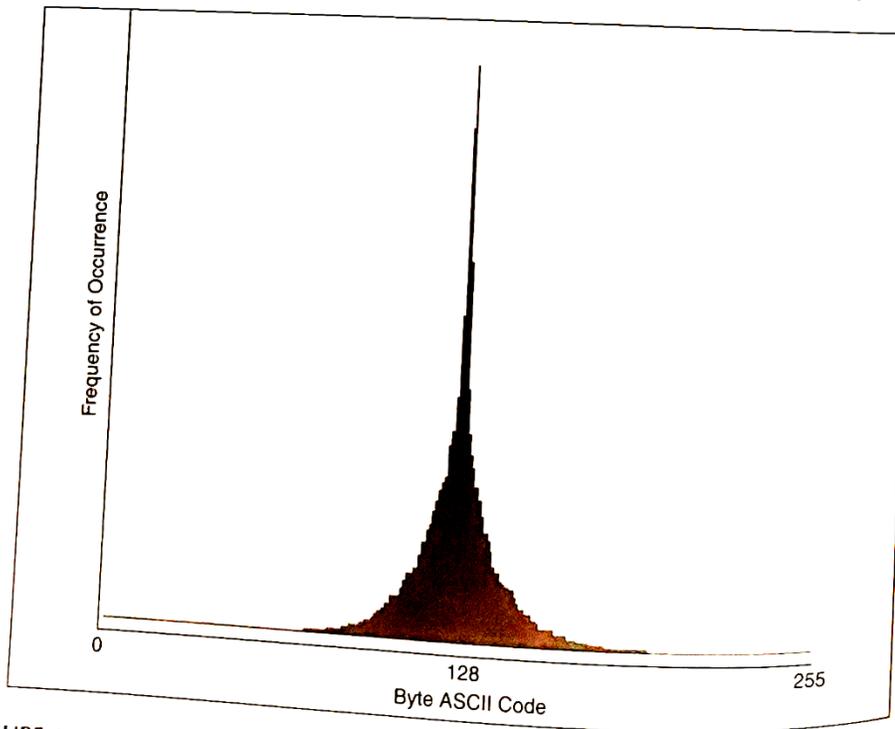

FIGURE 1

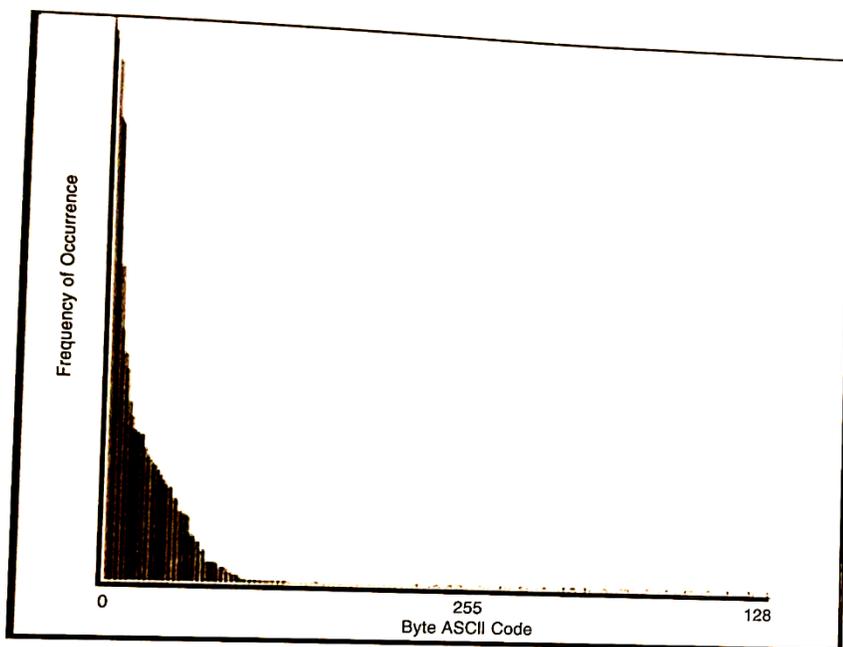Frequency distribution of ASCII bytes in a sound file of 66 Kbytes (11 kHz, 8 bit, mono).

FIGURE 2
Frequency distribution of ASCII bytes in a bitmap file with a size of 301 KB.
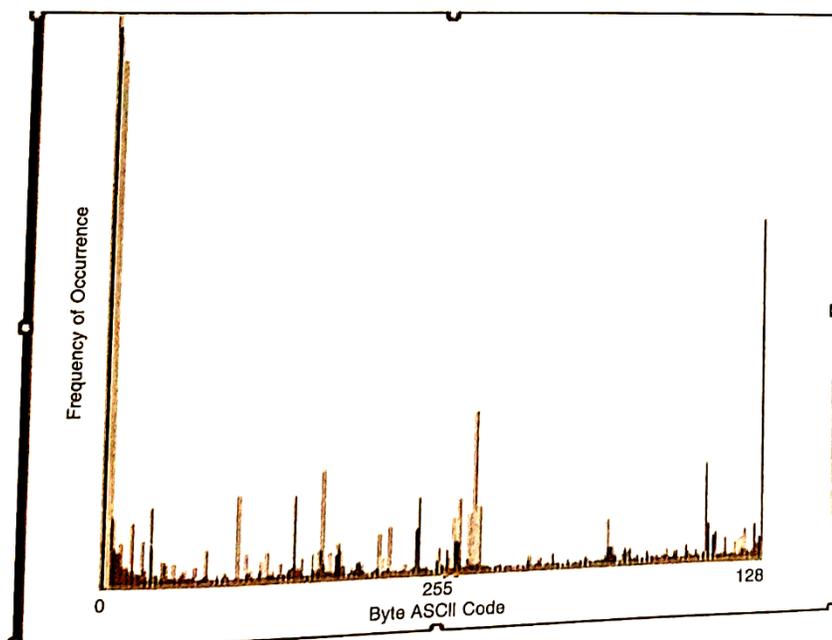

FIGURE 3
Frequency distribution of ASCII bytes in an EXE file with a size of 5.07 MB.

In this connection, we need to note that the byte is an artificial construct with an arbitrary (conscious) allocation of meaning to each group of eight bits. It is, therefore, further stated that: Consciousness is a prerequisite for the creation and perception of meaning in binary strings.

Bit Shift Distortion in Binary Strings

Since the proportion of zero and one bits in a meaningful string is no different from that of random strings, we decided to study the effect on meaning by changing the position of bits in strings.

Experiment 3—Meaning Degradation in Sound and Text Files

In this experiment, we shifted the position of a single bit from the beginning to the end of a data file and studied the effect of such a shift on the meaning of the file. Each target file was progressively changed by shifting the starting bit to the end, shifting the next bit to the end, and so on. To do this we removed the header from the target string, converted the bytes of the string to binary, shifted the relevant bit, changed the bit structure back to bytes, and replaced the header. The tests were performed on audio and text files and the results were checked for meaning by listening and reading for audio and text files, respectively.

Observations

It is intuitively obvious that a shifted bit will seriously degrade the meaning of a file. This is because once the first bit is shifted, every byte in the file will have a changed value. It may indeed be assumed that the entire content of the file will no longer be meaningful. We were surprised to find that this is not so.

In the case of audio files, we observed a steady degradation in the audio quality as bits were shifted. The degradation increases gradually and the file degrades into noise by the seventh bit shift. After this, when the eighth bit shift takes place, the entire original meaning returns since now, effectively, an entire byte has been shifted to the back of the file and the remaining byte sequence restored to its original condition.

We do not, as yet, have an explanation for this graceful degradation. Perhaps it is due to the fact that the silence byte (1111 1111), in the one-bit shifted condition will result either in 1111 1110 or 1111 1111, both of which are silent bytes. Since silence, ironically, is the principal component of audio files, a single bit shift does not render the file meaningless.

Table 2 shows the most prevalent byte in a sound file after 1 through 8 bit shifts. A clear pattern is evident from the table. This may indeed provide the explanation for why audio files retain meaning even when bytes are changed.

In the case of text files, a single bit shift completely destroys the meaning of the file. In view of the foregoing, we may hypothesize that the meaning of a string is more sensitively connected to its structure than to its contents or constituents.

Surprisingly, such results have been observed in chemical physics as well (Mathur and Mitra, 1979).

Headers—Strings to Look at Strings

Headers are strings that carry information about other strings. For example, the header of an audio string will carry information about its format, the sampling rate, whether it is

Table 2 The Effect of Bit Shifts on the Most Frequently Occurring Byte in an Audio File (11 kHz, 8 Bit, Mono)

No. of bits shifted	Most frequent byte	Hex value[a]	Binary value
1	254	FE	1111 1110
2	253	FD	1111 1101
3	251	FB	1111 1011
4	239	EF	1110 1111
6	223	DF	1011 1111
7	191	BF	1011 1111
8	127	7F	0111 1111

[a] The Hex value equals the base 16 value. In the base 16 representation of any number, the digits from 0 to 9 are represented by 0 to 9. The digits 10 to 15 are represented by A, B, ..., F in order. For example, the decimal number $127 = 7 \times 16 + 15$. So $127 = 7F$ in Hex notation.

mono or stereo, etc. Typically, headers are extremely small compared to the data strings they describe. However, they are extremely important to the meaning of strings as, without them, there would be no way to determine in what way the data needs to be processed to express meaning.

Experiment 4—Bit Pattern and Degradation in Sound Headers

In this experiment, we caused bit shift degradation in audio headers using the same methodology as for experiment 3.

Observations

All meaning is destroyed in a single bit shift. In fact, the data is no longer recognizable as sound at all. The data can no longer be expressed and, therefore, remains a string without meaning for as long as we like. However, with the original header, it can be played and, therefore, has a beginning and an end. It is tempting, though perhaps unwarranted, to compare the situation to an ancient verse that states: "That which is expressed (or has form) is temporal, that which is unexpressed (or has no form) is eternal." (*The Gita*, Hindu religious text of unknown authorship and chronology.)

Human Memory as Binary Strings

In our definition of meaning, we have postulated that memory is essential for the perception of meaning. In this context and in the context of the above experiments, it is important to examine the structure of human memory.

The Brain and Neural Networks

The human brain consists, for the most part, of a large network of neurons. Each neuron is essentially a biochemical electrical switch. Neurophysiology has not yet discovered by what biological mechanisms human memory is accomplished. Nor do we know, physiologically, how neurons store memories. We do know, thanks to computers, that memories can be stored in physical devices and that such processes as evoking, activating, and associating can be carried out with them (Simon, 1995, Braitenberg, 1984). Memories

in neural networks are stored in a repetitive and multiplexed fashion. As a result they show graceful degradation when bits are randomly changed in the network (Mitra, 1994).

If an artificial neural network can be constructed on a digital computer that has sufficient complexity to match the size of the human brain, it is possible that thought would result. While this is a controversial matter, for the purposes of this chapter, such a possible simulation seems to indicate the possibility that a brain can be represented as a binary string, much as an artificial neural network can.

Memory in Neural Networks

Artificial neural networks store data in several well-understood ways. The most common method is through the interconnection and weights of interconnections between neurons. Once again, the detail of the storage method is not important in our context. It is clear that memories can be stored as binary strings in other binary strings that represent neural networks.

DISCUSSION AND SPECULATION

Structure, Function, and Content

From the above experiments and analyses, it would appear that binary strings could be constructed out of anything and could represent any meaning if preceded by an appropriate header that can enable the string to be processed by an "expression engine" that expresses its meaning. The meaning is encrypted in the string through a conscious schema that can create an arbitrary meaning list for an arbitrary group of bits (8 in our case).

The meaning of strings would be dependent more on the structure and sequence of these groups of bits than on their specific content. The header string would entirely determine whether the string could be expressed or not. When expressed, a string would have a finite meaning.

Several questions remain unanswered in this work. Can a string have multiple meanings? Can strings be transmuted—audio to video, memory to meaning? In our future work we will attempt to answer some of these questions.

Strings, Meaning, and Consciousness

There would appear to be three functions that a string is capable of performing in the context of meaning. A string could represent a conscious memory that derives meaning from other strings that it interprets using appropriate header strings. Since all three functions are performed using strings that are indistinguishable from random strings, it is tempting to speculate that any string can have a meaning to a memory string through an appropriate header string.

Given this scenario, consider an n-dimensional matrix containing an infinite number of bipolar elements distributed randomly. Such a space would contain all possible forms of conscious memory, all possible forms of headers, and all possible meanings. In effect, the interplay of finite string lengths within this matrix would create all possible conscious realities. Whenever a string is expressed as a meaning, it would have a beginning and an end in an otherwise timeless situation.

ACKNOWLEDGMENTS

REFERENCES

Braitenberg, V., *Vehicles: Experiments in Synthetic Psychology,* Cambridge: MIT Press, 1984.

Feigenbaum, E. A., and Simon H. A., EPAM-like models of recognition and learning, *Cognitive Science,* 8:305–336, 1984.

Mathur, S.C. and Mitra, S., Crystal structure sensitivity of the band structure of organic semiconductors. *J. Phys. C Solid State*, 12, (2), 1979.

Mitra, S., The effect of synaptic disconnection on bi-directional associative recall. Proc. IEEE/SMC Conf., Vol. 1, 989, 1994.

Simon, H. A., Literary criticism: a cognitive approach, *SEHR*, Vol. 4, Issue 1, 1995.