How 'Unused' Codewords Make a Redundant Code

Paolo Rocchi
IBM
Via Shangai 53,
00144, Roma
paolorocchi@it.ibm.com

ABSTRACT
This paper tackles the theoretical questions connected to the calculation of various redundant forms in the information sector. In particular it is suggested a logical framework which reunifies the discussion of the active and the passive redundancies. We relate the present method to Shannon's entropy.

Categories and Subject Descriptors
H.1.1 **Systems and Information Theory** (E.4) Data Structures and Data Storage Representations.

General Terms
Measurement, Design, Reliability, Standardization, Theory.

Keywords
Coding theory, passive redundancy, active redundancy, entropy.

1. INTRODUCTION
From Shannon's perspective, redundancy is the presence of more symbols in a message than is necessary. This interpretation leads to the following definition of redundancy [1]

$$R = \frac{H_{\max} - H_{act}}{H_{\max}}$$

Where *actual entropy* quantifies the source in use, and *maximum entropy* is when all the symbols of the source have equal probability.
It is necessary to underline how Shannon develops his theory with a very specific model in mind. He postulates an ensemble of messages to be transmitted whose statistics were fully known and unchanging. These are to be passed down a channel whose properties are accurately known. Also he assumes that the messages can be subdivided into blocks as required.
This model turns out to be insufficient to study redundancy in many modern sophisticate systems e.g. multimodal systems [2], neural networks [3], linguistics [4], etc. To exemplify linguists analyze the *stylistic redundancy* which relies on subjective feeling and the *grammatical redundancy* due to the objective rules of each language. In conclusion redundancy can take so many different shapes that having a single model and criterion for measure it may be a little misleading.
In particular Shannon's theory does not clarify how a sole code can exploit dissimilar forms of redundancy. We illustrate this problem emerging in the digital technologies through the following case.
Take the set of decimal numbers from 0000 up to 9999. We assume the code *A* includes the first hundred codewords while the remaining words are unused

$A = \{0000 \div 0099\}$	used
$B = \{0100 \div 9999\}$	unused

If the receiver gets an unused codeword, he detects an error, and this check brings evidence how *A* is redundant. Now we consider again the numbers from 0000 to 9999 but define a different four-figure code. Let *C* codeword has two decimal figures followed by the check-sum of the first pair. The ensuing table sums up the used codewords of *C* when the remnants make the set *D*

$C = \{0000, 0101, 0202,...9918\}$	used
$D = \{0001,0002...0100,0102,...9999\}$	unused

The code *C* exhibits a redundancy similar to the redundancy of *A* due to 9900 unused codewords. In addition we find that each codeword of *C* is redundant due to the checksum digits. The receiver may use two algorithms to discover a corrupted word. He finds an error:

1. When the received codeword belongs to *D*.
2. When the first digits of the received codeword mismatch with the checksum digits.

This couple of algorithms brings evidence that *C* includes two forms of redundancy.

Now we calculate the redundancy of A. As A uses the decimal base, the maximum entropy is

$$H_{max}(A) = - \log 1/10 = 2.302 \text{ bits/figure}$$

To obtain the actual entropy we follow the approximate method suggested by Shannon. We calculate the entropy of one hundred codewords of A (assuming equally likely) and divide the result by the number of figures of each codeword

$$H_{act}(A) = (- \log 1/100)/4 = 4.605/4 = 1.151 \text{ bits/figure}$$

Thus redundancy of A ranges

$$R(A) = \frac{2.302 - 1.151}{2.302} = 0.5 \text{ bits/figure}$$

Assuming the codewords of C with the statistics like that of A for the sake of simpleness, we have $H_{max}(C)=H_{max}(A)$ and $H_{act}(C)=H_{act}(A)$ and thus we obtain

$$R(C) = R(A)$$

The theoretical redundancy of C equals to the theoretical redundancy of A, but this unique value corresponds badly to the physical reality because C shows two forms of redundancy in the practice and A has one form of redundancy. Hence we should clarify: Does $R(C)$ quantify algorithm 1, algorithm 2 or neither? How can we calculate every form of redundancy emerging from coding?

Modern authors discuss various forms of redundancy preferably through qualitative discourses [5] instead the present paper puts forward a method of calculus to reply the foregoing questions. In particular the paper imports the notions of *passive redundancy* and *active redundancy* from the reliability theory. Last it will be shown how the results consist with the Shannon theory.

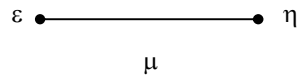
2. SYSTEMS AND MODULES

Let the generic component ε_i executes the function μ_i (i=1,..n); we define the system S as the finite set of n pairs

$$S = [(\varepsilon_1, \mu_1), (\varepsilon_2, \mu_2), (\varepsilon_3, \mu_3), ... (\varepsilon_n, \mu_n)] \qquad (1)$$

This expression easily interprets a productive structure, such as a machine, an assembly line etc. In fact the generic module (ε_i, μ_i) is the operational unit that brings forth the function μ_i. The introduction of (1) in the information territory is more delicate and requires some details.

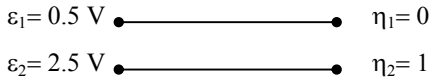
Technical writers usually assume that a signal is a physical quantity. They share the idea that information has a material origin and has not an ethereal nature. As second, boundless literature shares the idea that an item of information represents something [6][7]. We formalize this couple of assumptions in the following manner. We establish that an item of information (e.g. a message, a codeword, a signal) is the algebraic entity ε which works as the model of the object η. The following semantic diagram formalizes ε that symbolizes η, namely the piece of information ε brings forth the semantic function μ



(2)

Hence we define the pair (ε,μ) when the informational item ε executes the semantic activity μ.

Example A: Take a TTL circuit with low signal 0.5 volt and high signal 2.5 volt. This semantic diagram shows how the voltage values mean respectively the bits 0 and 1



$$S_A = [(\varepsilon_1, \mu_1), (\varepsilon_2, \mu_2)]$$

Example B: A congress web page includes four sections illustrating 'program committee', 'congress program', 'call for paper', and 'tutorial'. The whole communication system S_B should be formalized in the following terms

$$S_B = [(\varepsilon_1, \mu_1), (\varepsilon_2, \mu_2), (\varepsilon_3, \mu_3), (\varepsilon_4, \mu_4)]$$

3. A DEFINITION OF REDUNDANCY

In general usage, the term 'redundancy' signifies more of anything than is strictly needed, usually resulting from repetition or duplication. Both the repetition of information and the inclusion of extra information so as to reduce errors in understanding messages are considered redundant. We translate this property into the following analytical determination of the redundancy for S.

We assume one module is necessary to fulfill μ_j, and e_j modules of S ($n \geq e_j > 1$) accomplish the operation μ_j; thus the surplus modules ($e_j - 1$) provides the *redundancy* r_j of the function μ_j. To generalize the present conceptualization, we assume e modules of S perform m functions $\mu_1, \mu_2, ..., \mu_m$ ($n \geq e \geq m$). As redundancy means abundance, the summation of the partial redundancies yields the redundancy of the whole system.

Definition

$$r = \sum_{j}^{m} r_j = \sum_{j}^{m} (e_j - 1) = (e - m) \qquad (3)$$

If summation is zero the system is not redundant; if at least one partial redundancy is positive, the system *S* is redundant.
Definition (3) may be applied to different areas as a characteristic of generality of the present study [8]. We turn our attention to the information territory where Definition (3) can calculate a broad variety of applications. In fact, whenever two or more pieces of information convey the same content, this information is redundant.

Example C: Twenty-five large advertising posters plaster the fronts of the houses in a square. They deliver the same message the redundancy of this communication system is greater than zero

$$r_C = (25 - 1) = 24 \text{ messages}$$

Definition (3) can also calculate involute meanings. Take for example a codeword that carries on a value which the receiver extracts through the calculus. Eqn (3) quantifies the redundancy caused by hidden information.

Example D: The code *C*, discussed in the Introduction of the present paper, includes the following codeword with two checksum digits located on the far right

99 18

The receiver manages the following numbers: ε_1=99, ε_2=18, and the number ε_3 = (9+9) = 18. In fact the leftmost part of the message conveys the value 18 whose significance derives from calculation and is not explicit. The codeword has the following structure

$$S_D = [(\varepsilon_1, \mu_1), (\varepsilon_2, \mu_2), (\varepsilon_3, \mu_2)]$$

Using (3) we obtain

$$r_D = (2 - 1) = 1 \text{ number}$$

The codeword is redundant due to the pairs (ε_2, μ_2), (ε_3, μ_2) having the same significance. This calculation clarifies how algorithm 2 can detect errors thanks to this property.

Eqn. (3) provides the redundancy in a deterministic manner, next we shall correlate this approach to the Shannon probabilistic method.

4. RESERVE CODEWORDS

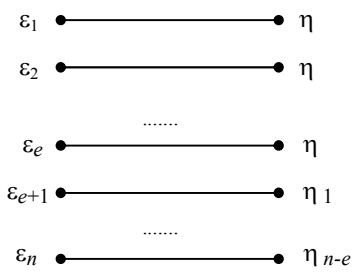
Doubled components make a fabric more robust and engineers design a redundant machine to enhance the machine reliability [9]. There are two main types of redundancy in engineering: active and passive. The active redundancy is the most expensive to implement. This configuration features two or more active units carrying the same job, with only one set connected to the outputs. If a fault occurs, there is automatic, loss-less support of the remaining units. Passive redundancy is the cheapest to implement. One or more backup units are installed in addition to one active unit. When a failure occurs, an automatic switch starts a reserve unit and the system does not interrupt.

We discuss the active and the passive redundancy for digital codes.

I) - We consider *n* codewords obtained by means of the base *B* ($B \geq 2$) and the fixed length L_{act}

$$n = \left(B^{L_{act}} \right) \qquad (4)$$

Now we assume the code *S* has *n* words (4). Let *e* codewords of *S* signify one sole object η ($n > e > 1$) and the remaining codewords stand for ($n - e$) objects. The semantic diagram exhibits this coding



(5)

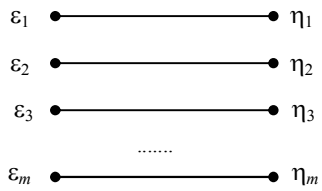
We obtain the redundancy of *S* from (3)

$$r = \sum_{j}^{m} r_j = (e - 1) > 0 \qquad (6)$$

This result gauges the *active* redundancy of code (5) because all the codewords are in use. Examples C and D calculate further cases of active redundancy.

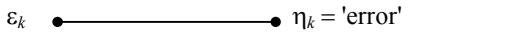
II) - Now we assume *S* has *m* codewords ($n > m$) that signify *m* objects $\eta_1, \eta_2, \ldots \eta_m$ and the remaining ($n - m$) codewords are 'unused'. The semantic diagram shows *m*

codewords running while the remnant codewords ε_{m+1} , ε_{m+2} ,...., ε_n do not compare in the diagram



(7)

We detail the role of the words ε_{m+1} , ε_{m+2} ,...., ε_n from a practical perspective.
Frequently a word out of the set $\{\varepsilon_{m+1}$, ε_{m+2} ,...., $\varepsilon_n\}$ is transmitted by mistake, hence this word leaves the 'unused' state and becomes active. That word tells 'error' (or something like) to the receiver and the semantic diagram formalizes this semantic activity



$k = m+1, m+2,... n$

(8)

A word belonging to the group $\{\varepsilon_{m+1}$, ε_{m+2} ,...., $\varepsilon_n\}$ turns out to be a stand-by word, because a word, transmitted and recognized as a 'wrong' word, executes a semantic function. Modern theorists keep separated the 'used' codewords of a coding from the 'unused' codewords, as if the latter will be definitively kept aside. The 'used' and 'unused' codewords lie apart on the theoretical plane, instead all of them may be transmitted, stored, manipulated etc. in the practice. Everyday experience brings evidence how the 'unused' codewords are used in the reality and convey information. In other terms $(n - m)$ codewords of code (7) act as reserve words because they occasionally get out the stand-by state and exhibit a message. This means that encoding (7) embodies the *passive redundancy* in the information field.
Passive redundancy consist of codewords busy and other codewords at disposal, we conclude that all the codewords are available for the semantic functions, and e equals to n in (3)

$$e = n \tag{9}$$

And the passive redundancy is

$$r = \sum_{j}^{m} r_j = (n - m) \tag{10}$$

The diagrams (7) bring evidence that the set $\{\varepsilon_1$, ε_2 ,...., $\varepsilon_m\}$ is not redundant and (10) shows that the redundancy of (7) is caused exclusively by the stand-by words that are $(n - m)$. Expression (9) has relevant significance because it quantifies the passive redundancy and specifies how the passive redundancy relies on stand-by codewords.

5. TWO FORMS OF REDUNDANCY

The control of the received words turns out to be demanding and the designers of a code frequently want the words be easily recognized. They add some controlled digits to a word to improve the chances of being able to verify and to recover the original message. In substance they combine *the passive redundancy of the codeword with the active redundancy of the whole encoding*.

Example E: We complete the calculation of C undertook in Example D, in detail the quantity r_D provides the active redundancy of a word which enables the control algorithm 2

$$r_D = 1 \text{ number} = r_{E1}$$

Moreover 9900 spare codewords allow the control accomplished by algorithm 1 due to the following passive redundancy

$$r_{E2} = n - m = 10000 - 100 = 9900 \text{ codewords}$$

The quantities r_{E1} and r_{E2} show how C has two distinct forms of redundancy.

Anyhow we select one hundred codewords out of the set $\{0000 \div 9999\}$ to convey information, the passive redundancy does not vary because r_{E2} relies on the amount of surplus codewords. This passive redundancy is equal for whichever set of used codewords, hence r_{E2} provides the redundancy of the code A too.

The passive redundancy of the overall encoding and the active redundancy of the single word absolutely differ from the practical and the theoretical viewpoints. In particular the active redundancy of the single codeword does not raise special difficulties; the calculus is rather easy. Conversely the passive redundancy of a code requires further developments.

6. FURTHER NOTES ON THE PASSIVE REDUNDANCY

Let L_{act} the actual fixed length of the passive-redundant code S and L_m is the fixed length of the minimal encoding. By definition, the minimal code has the length just necessary to symbolize m objects η. We make explicit (10) by the combinatory analysis

$$r = (n - m) = (B^{L_{act}} - B^{L_m}) \quad (11)$$

The base B is not lower than 2, hence (11) entails that S is redundant if and only if L_{act} is larger than the minimal length. The code S is redundant whenever L_{act} surpasses L_m

$$r \geq 0 \quad \Leftrightarrow \quad (L_{act} - L_m) \geq 0 \qquad B \geq 2 \quad (12)$$

This fundamental expression leads authors to appreciate the passive-redundancy of S by means of the relative increase of length with respect to the minimal length [10]. The difference of length is the simplest pragmatic assessment of the passive redundancy for digital encoding

$$D(S) = L_{act} - L_m \quad (13)$$

Several technical authors introduced a measurement symmetrical to $D(S)$ but they have not given the logical account of such a measurement so far. Instead the passive redundancy and in particular Eqn (12) justify the origin of (13).

Eqn (12) shows how the calculation of the minimal length is the basic reference for the definition of the passive redundancy. Combinatory analysis provides this equation useful for crude evaluations of L_m

$$L_m = \log_B m \quad (14)$$

Example F: We use Eqn (14) to calculate the number of bits necessary to encode the decimal values from 00 to 99

$$L_m = \log_2 100 = 6.64 \text{ bits}$$

Engineers round L_{act} to seven bits because L_{act} is not an integer, and construct out a passive redundant code which ranges

$$D(S)_F = L_{act} - L_m = 7.00 - 6.64 = 0.36 \text{ bits} > 0$$

7. FIRST SHANNON'S CONTRIBUTION

The lower bound of the word length arises as the central problem to calculate the passive redundancy. The length (14) is valid when we overlook the frequency of the signals, although statistical distribution of the modules causes significant consequences.

Shannon starts from a system model more accurate than (1) since he takes the triad (ε_i, μ_i, p_i) instead of the pair (ε_i, μ_i). The probabilities p_i verifies the following constraint

$$\sum_i^n p_i = 1 \quad (15)$$

Shannon quantifies the average number of symbols necessary to represent the set $\{\eta_1, \eta_2, ...\eta_m\}$ on the basis of the entropy function H, and calls L_o *optimal* or *minimal length with assigned statistical distribution* the following expression

$$L_o = H = -k\sum_i^m p_i \log_B p_i \qquad k > 0 \quad (16)$$

When the probability is constant

$$p_i = 1/m \qquad i = 1, 2.m \quad (17)$$

The function entropy is maximum and equals to (14)

$$\max L_o = H_{\max} = \sum_i^m 1/m \log_B m = \log_B m = L_m \quad (18)$$

Notably, the optimal length L_o is not greater than L_m

$$L_o \leq L_m \quad (19)$$

Now we can redefine the distance (13) using L_o and this new parameter should be used when we consider the statistical distribution of the words

$$D'(S) = L_{act} - L_o \quad (20)$$

In short, the present theory calculates the minimal length and in turn the passive redundancy at two levels of accuracy. We have the lower bound L_m if we ignore statistical distribution of signals of S; we obtain the lower bound L_o if we calculate the statistical distribution using Shannon's method. Eqn. (18) proves how the two methods are consistent.

8. SECOND SHANNON'S CONTRIBUTION

Shannon calculates the channel capacity, a bound on the maximum amount of error-free digital data which can be transmitted over a communication link in the presence of the noise interference. This author is more interested on the flow of codewords rather than on the words one by one.

Shannon establishes that an information source produces a sequence of signals to be communicated to the receiving terminal and this sequence makes a stochastic system. Thus he assumes the system S, say '*source*

alphabet', generates the discrete Markov chain S_T called '*source text*'. In brief the pairs of *S* make a chain by time passing.
Normally the source text has repeated patterns of symbols because the symbols are not absolutely random. E.g. in English the letter 'q' must always be followed by a 'u'. This fact leads Shannon to calculate the redundancy of S_T through the actual entropy of *S*, and the maximum entropy (20) that marks the equally likely distribution of symbols

$$R = \frac{L_m - L_{act}}{L_m} = \frac{H_{\max} - H_{act}}{H_{\max}} \qquad (21)$$

This quantity is always less than the unit because of (19) and (12).

It is necessary to underline that (21) quantifies the redundancy of the data stream S_T, instead Definition (3) applies to any form of redundancy.

In conclusion the present logical framework which starts with a deterministic definition of redundancy covers also the probabilistic study of redundancy and consists with Shannon' conceptualization which focuses on a particular aspect of redundancy.

9. CONCLUSION

Modern information theory calculates a sole form of redundancy whereas experience shows a variety of redundant forms emerging in the digital technology. As a consequence practitioners do not obtain the full degree of accuracy and completeness when they assess a redundant solution.
This paper derives Definition (3) from the idea of abundance and infers a number of measurements for it.
In particular we have discussed two kinds of redundancies, named *active* and *passive*. The paper clarifies that both types of redundancies normally coexist in a sole code, and relates them to Shannon's information theoretic treatment.
The author provides examples to calculate the redundancy of a single word, of an encoding and of a sequence of transmitted words through the entropy.

10. REFERENCES

[1] C.E. Shannon, A mathematical theory of communication, *Bell Syst. Tech. J.* , 27, 1948, 379-423.
[2] K. McKenzie Mills, J.L. Alty, Integrating speech and two-dimensional gesture input: A study of redundancy between modes, *Proc. Conf. Computer Human Interaction*, Australasian, 1998, 6-13.
[3] H.Barlow, Redundancy reduction revisited, *Network*, *12*(3), 2001, 241-53.
[4] M. Gillette, E.J. Wit, What is linguistic redundancy? ,http://galton.uchicago.edu/～ wit/redundan.html,1998.
[5] P. Horwich, Meaning, use, and truth, *Mind*, 1995, *204*(414), 355-368.
[6] P. Cobley, R. Appignanesi, *Introducing semiotics*, Victoria, Totem Books, 2001.
[7] A.W. Moore, *Meaning and reference*, Oxford, Oxford Univ Press, 1993.
[8] P.Rocchi, W.Betori, A Unified way to calculate redundant resources, *Proc. Conf. on Computational Intelligence for Modelling Control and Automation*, Vienna, 2005, *2*, 396-401.
[9] E.E. Lewis, *Introduction to reliability engineering*, N.Y. Wiley, 1995.
[10] P.Rocchi, Some notes for the uniform calculus of redundancy, *Proc. IEEE Conf. on Systems, Man and Cybernetics (SMC)*, The Hague, 2005, 1649-1653