

Efficient Data Representation

34

The Mathematical Theory of Communication

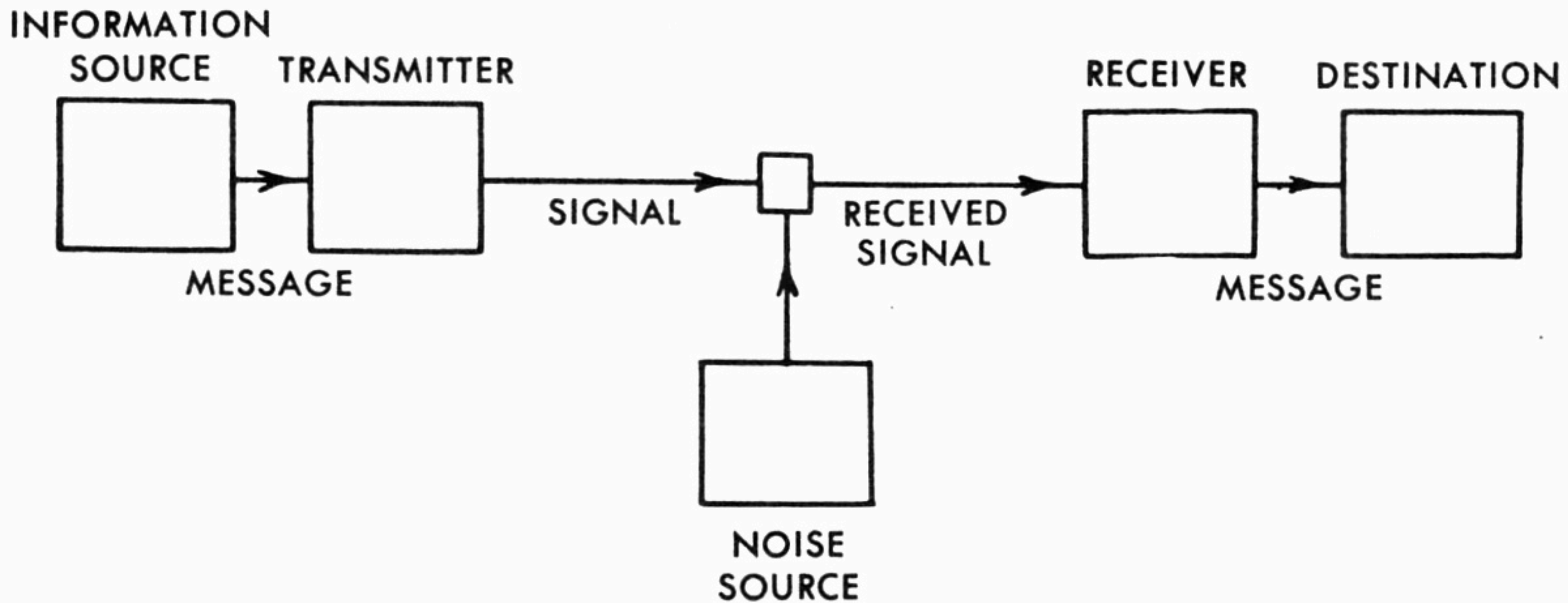


Fig. 1. — Schematic diagram of a general communication system.

Brief Recap

Parity bit: a simple method to check errors in data

Hash function: converts the input into a value of fixed length hash code

*Collisions are when the same inputs generate the same hash codes

Cryptographic hash codes are also designed to be computationally infeasible to reverse

Check sum - designed to detect the most common errors in the data and often to be fast to compute

Claude Shannon - Information Theory



The Bell System Technical Journal

Vol. XXVII

July, 1948

No. 3

A Mathematical Theory of Communication

By C. E. SHANNON

INTRODUCTION

THE recent development of various methods of modulation such as PCM and PPM which exchange bandwidth for signal-to-noise ratio has intensified the interest in a general theory of communication. A basis for such a theory is contained in the important papers of Nyquist¹ and Hartley² on this subject. In the present paper we will extend the theory to include a number of new factors, in particular the effect of noise in the channel, and the savings possible due to the statistical structure of the original message

Entropy in one sentence

Information entropy is the **average** rate at which **information** is produced by a data source.

Entropy part 2

Low entropy (e.g., 0) = predictable

Entropy = 0 when an outcome is certain

High entropy (e.g., 1) = surprise

Entropy = 1 when an outcome is evenly balanced

Entropy is defined by probabilities. The more uncertain or random is in a data source, the more information it will contain.

Information Entropy

Fair coin toss: probability is 50-50 then the entropy is at it's highest: 1



A double sided coin toss: probability is 100-0, the entropy is at it's lowest: 0



Entropy effectively bounds the performance of the strongest lossless compression possible.

The limit of compression will always be the entropy of the message source



Information Theory

How many bits required to communicate:

the result of a coin toss?



the result of a throw of a die?



the throwing of a six?



Compression

Exploit runs of 0s to transmit in less than 1 bit.

Compression

Encode information using fewer bits than original representation OR to squeeze out unnecessary bits.

The higher the entropy the less we can compress

The lower the entropy the more we can compress

We can NOT compress beyond the entropy limit without throwing some data away

Lossy: JPEG, MPEG

Lossless: Zip

https://en.wikipedia.org/wiki/Huffman_coding

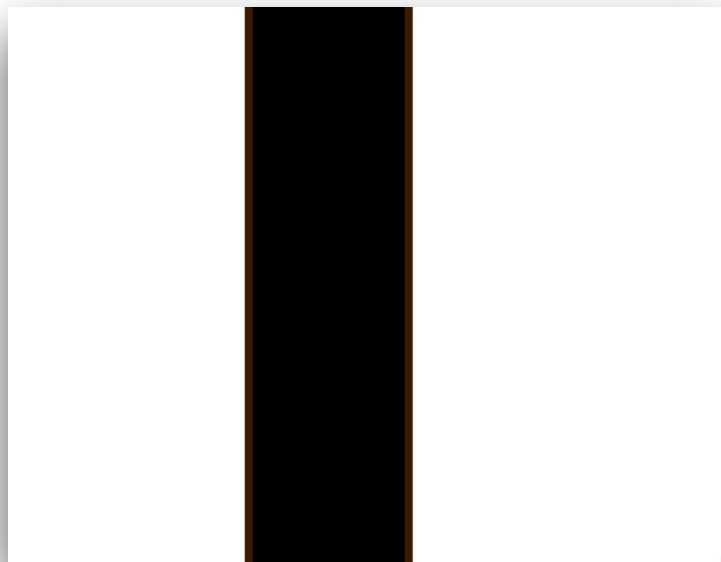
But why would we want to compress?

Lossless Compression

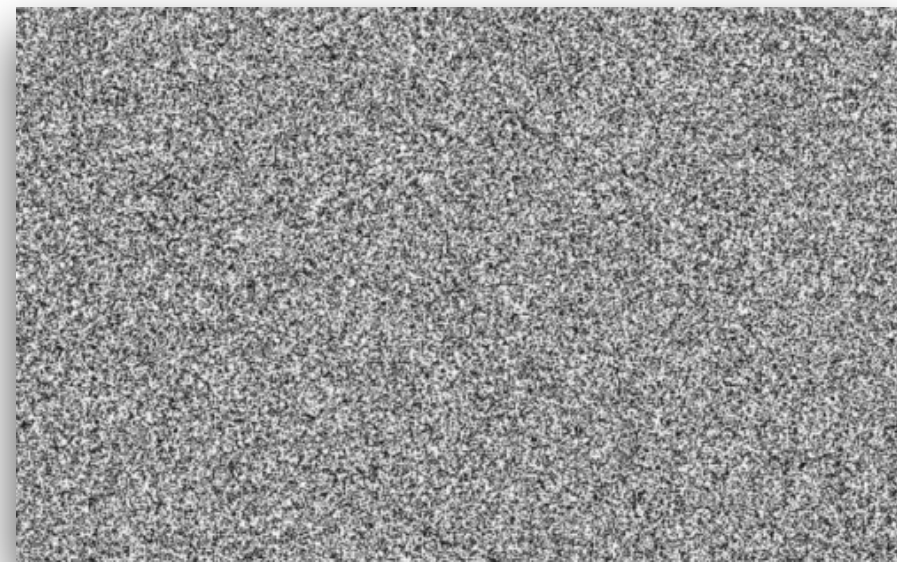
Lossless: Zip

[https://en.wikipedia.org/wiki/Zip_\(file_format\)](https://en.wikipedia.org/wiki/Zip_(file_format))





1 Bar.tiff



2 WhiteNoise.tiff



Which image will be more compressible (without loss)?

 1 Bar.tiff	Today, 10:56	6 KB
 1 Bar.tiff.zip	Today, 11:36	2 KB
 2 WhiteNoise.tiff	Today, 10:54	319 KB
 2 WhiteNoise.tiff.zip	Today, 11:36	317 KB

COMP 30660 Computer Architecture & Organisation

Consider:

Represent this as:

Requires a probability imbalance to be effective
i.e lowish entropy

Predictability = compressible

Lossy Compression - jpg



Huffman Codes

1098

PROCEEDINGS OF THE I.R.E.

September

A Method for the Construction of Minimum-Redundancy Codes*

DAVID A. HUFFMAN[†], ASSOCIATE, IRE

Summary—An optimum method of coding an ensemble of messages consisting of a finite number of members is developed. A minimum-redundancy code is one constructed in such a way that the average number of coding digits per message is minimized.

INTRODUCTION

ONE IMPORTANT METHOD of transmitting messages is to transmit in their place sequences of symbols. If there are more messages which might be sent than there are kinds of symbols available, then some of the messages must use more than one symbol. If it is assumed that each symbol requires the same time for transmission, then the time for transmission (length) of a message is directly proportional to the number of symbols associated with it. In this paper, the symbol or sequence of symbols associated with a given message will be called the "message code." The entire number of messages which might be transmitted will be called the "message ensemble." The mutual agreement between the transmitter and the receiver about the meaning of the code for each message of the ensemble will be called the "ensemble code."

Probably the most familiar ensemble code was stated in the phrase "one if by land and two if by sea." In this case, the message ensemble consisted of the two individual messages "by land" and "by sea", and the message codes were "one" and "two."

In order to formalize the requirements of an ensemble code, the coding symbols will be represented by numbers. Thus, if there are D different types of symbols to be used in coding, they will be represented by the digits $0, 1, 2, \dots, (D-1)$. For example, a ternary code will be constructed using the three digits 0, 1, and 2 as coding symbols.

The number of messages in the ensemble will be called N . Let $P(i)$ be the probability of the i th message. Then

$$\sum_{i=1}^N P(i) = 1. \quad (1)$$

The length of a message, $L(i)$, is the number of coding digits assigned to it. Therefore, the average message length is

$$L_{av} = \sum_{i=1}^N P(i)L(i). \quad (2)$$

The term "redundancy" has been defined by Shannon¹ as a property of codes. A "minimum-redundancy code"

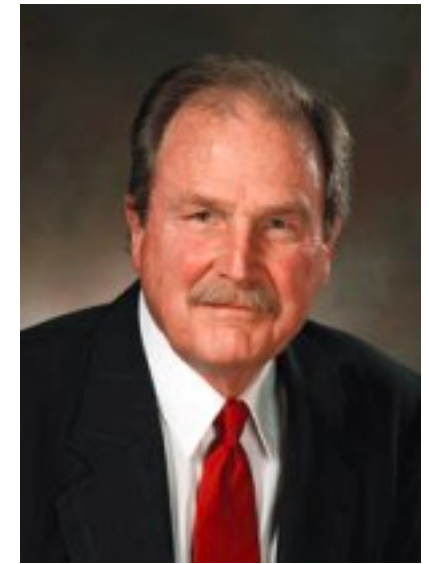
will be defined here as an ensemble code which, for a message ensemble consisting of a finite number of members, N , and for a given number of coding digits, D , yields the lowest possible average message length. In order to avoid the use of the lengthy term "minimum-redundancy," this term will be replaced here by "optimum." It will be understood then that, in this paper, "optimum code" means "minimum-redundancy code."

The following basic restrictions will be imposed on an ensemble code:

- (a) No two messages will consist of identical arrangements of coding digits.
- (b) The message codes will be constructed in such a way that no additional indication is necessary to specify where a message code begins and ends once the starting point of a sequence of messages is known.

Restriction (b) necessitates that no message be coded in such a way that its code appears, digit for digit, as the first part of any message code of greater length. Thus, 01, 102, 111, and 202 are valid message codes for an ensemble of four members. For instance, a sequence of these messages 111102202010111102 can be broken up into the individual messages 111-102-202-01-01-111-102. All the receiver need know is the ensemble code. However, if the ensemble has individual message codes including 11, 111, 102, and 02, then when a message sequence starts with the digits 11, it is not immediately certain whether the message 11 has been received or whether it is only the first two digits of the message 111. Moreover, even if the sequence turns out to be 11102, it is still not certain whether 111-02 or 11-102 was transmitted. In this example, change of one of the two message codes 111 or 11 is indicated.

C. E. Shannon¹ and R. M. Fano² have developed ensemble coding procedures for the purpose of proving that the average number of binary digits required per message approaches from above the average amount of information per message. Their coding procedures are not optimum, but approach the optimum behavior when N approaches infinity. Some work has been done by Kraft³ toward deriving a coding method which gives an average code length as close as possible to the ideal when the ensemble contains a finite number of members. However, up to the present time, no definite procedure has been suggested for the construction of such a code



David A. Huffman
1952

* Decimal classification: R531.1. Original manuscript received by

² R. M. Fano, "The Transmission of Information," Technical Report No. 65, Research Laboratory of Electronics, M.I.T., Cam-

Weather App - Efficient Encoding

75%



01

10%



10

10%



11

5%

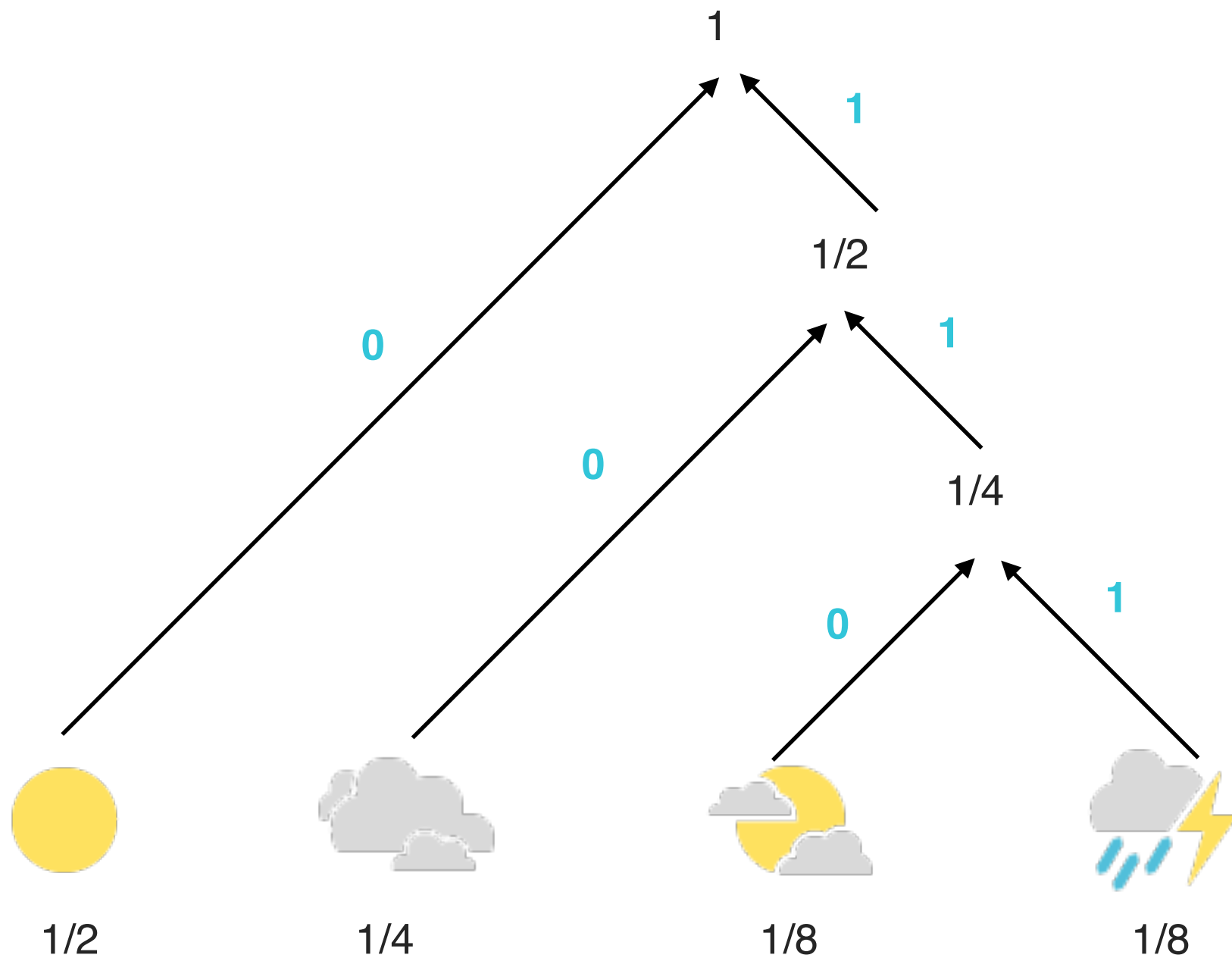


00

We can use the underlying probability of the events to create more efficient communication codes.



Weather codes



Huffman Coding

Event	Freq	Code
Sunny	$1/2$	
Cloudy	$1/4$	
Mixed	$1/8$	
Stormy	$1/8$	

Building Huffman Trees

Low frequency chars should be far from the root
i.e. longer codes

1. Each character is assigned a leaf node
2. Select the two lowest frequency character nodes
 1. Link with an internal node and sum their frequencies to get the frequency for the internal node.
3. Repeat from 2 till there is just one node

Ties:

Resolve ties randomly

Sometimes many equally good trees

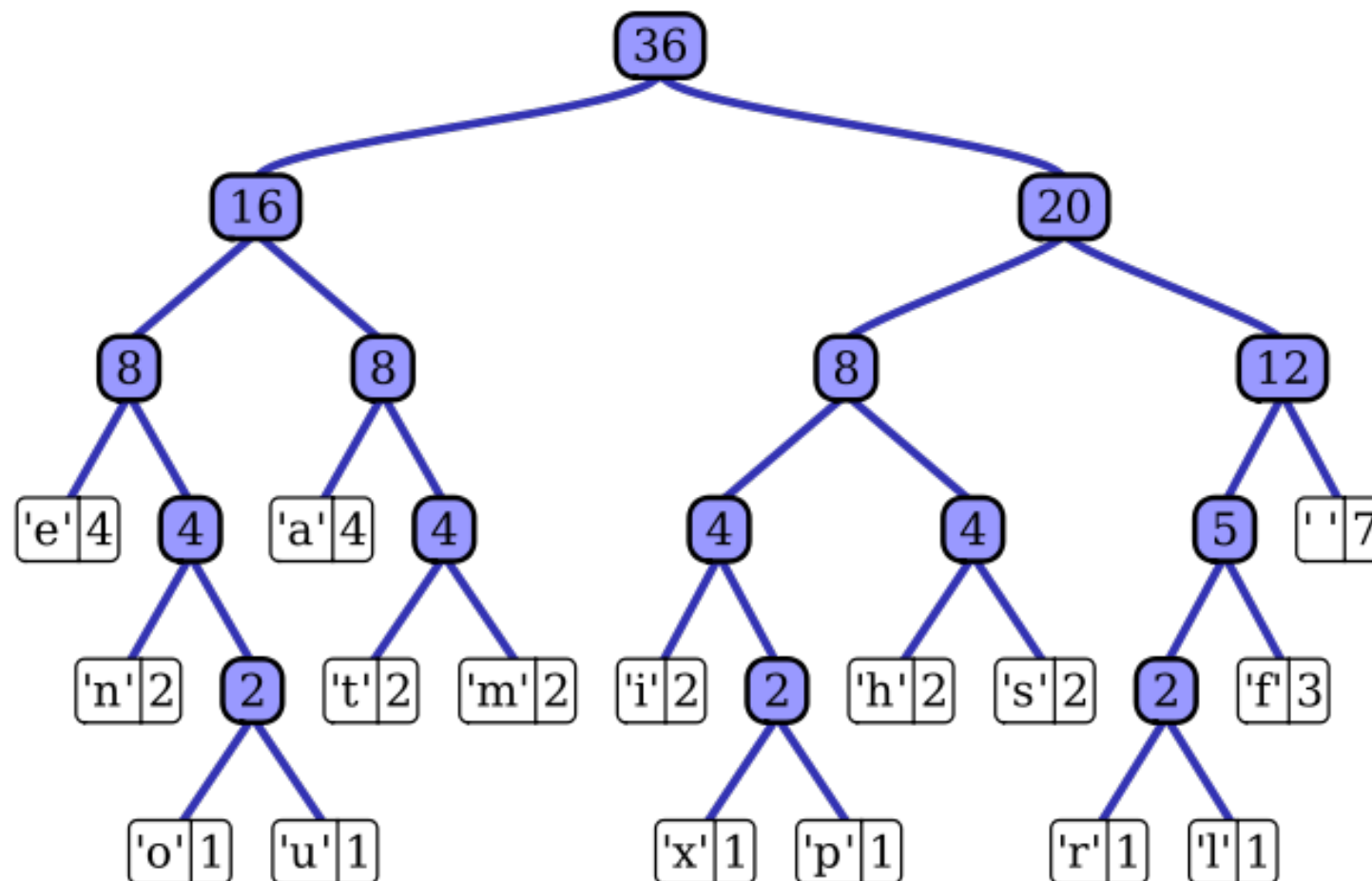
One n -character code is as good as another n -character code

Compression

Huffman Coding

Sample text:

- ▶ “this is an example of a huffman tree”



Huffman Coding

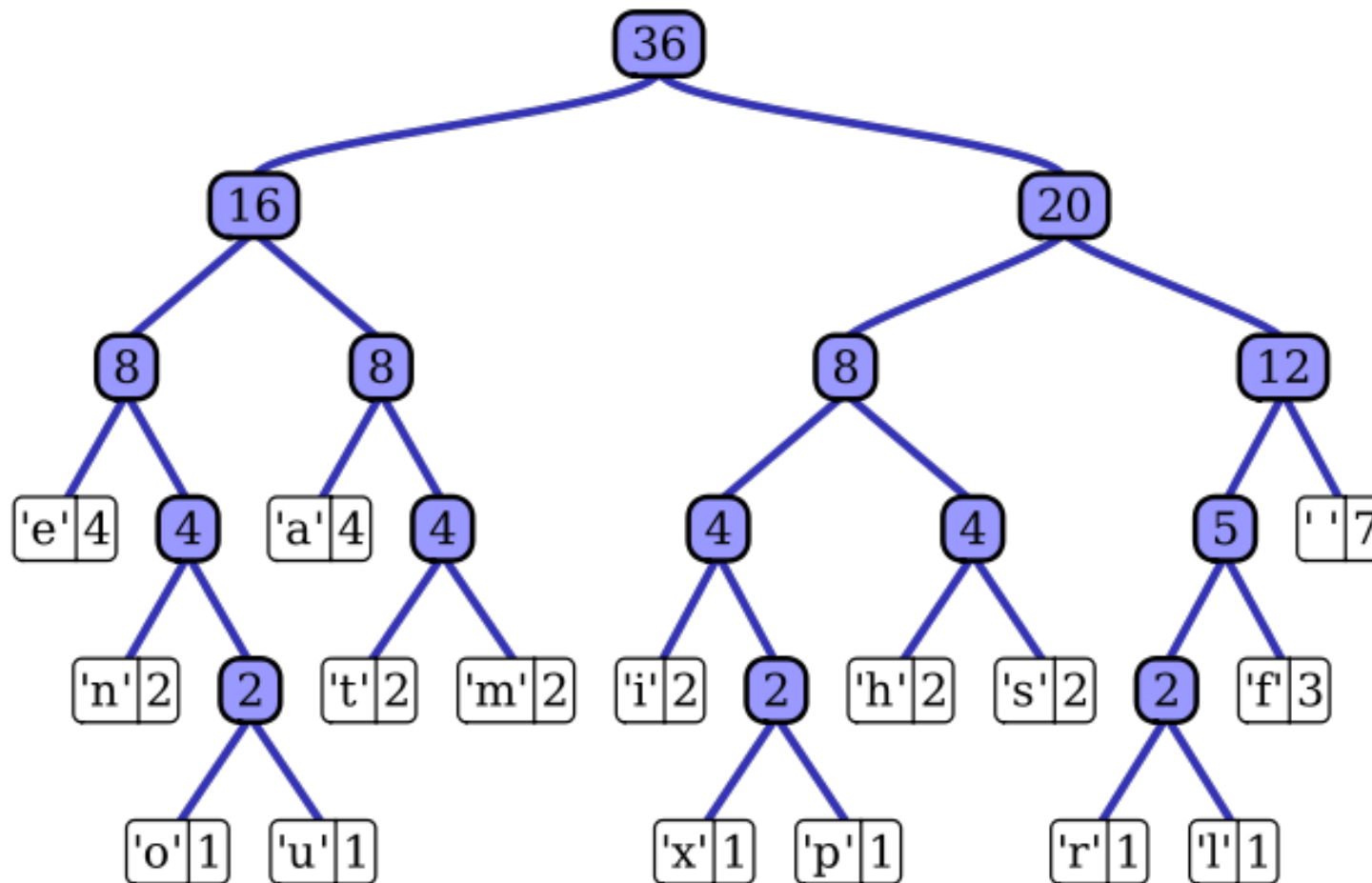
Char	Freq	Code
space	7	
a	4	
e	4	
f	3	
h	2	
i	2	
m	2	
n	2	
s	2	
t	2	
l	1	
o	1	
p	1	
r	1	
u	1	
x	1	

Compression

How does the tree work?

What is?

011100110001100010111
10111010001100110



Huffman Coding

Char	Freq	Code
space	7	111
a	4	010
e	4	000
f	3	1101
h	2	1010
i	2	1000
m	2	0111
n	2	0010
s	2	1011
t	2	0110
l	1	11001
o	1	00110
p	1	10011
r	1	11000
u	1	00111
x	1	10010