Homework Assignment #1

Source Encoding: NY Times Article

By

Zuoda Ren

Zhening Li

Zeqi Chen

Weimin Zhou

Faculty Advisor:

Dr. Paul S. Min (Electrical and System Engineering)

ESE471

Department of Electrical and System Engineering

Washington University

Saint Louis, Missouri

Feb 10, 2016

Methods

After several days' discussion and research, we decide to use Huffman encoding algorithm to transmit this article. A Huffman code can be represented as a binary tree whose leaves are the symbols that are encoded. At each non-leaf node of the tree there is a set containing all the symbols in the leaves that lie below the node. In addition, each symbol at a leaf is assigned a weight (which is its relative frequency), and each non-leaf node contains a weight that is the sum of all the weights of the leaves lying below it. The weights are not used in the encoding process. In this methods, more common symbols are generally represented using fewer bits than less common symbols.

Our work can be roughly divided into four parts.

Part 1: Counting the frequency of every symbol and computing the appearance probability of each symbol. (See attachment #1)

Part 2: Encoding every symbol using Huffman encoding algorithm in Matlab. (See attachment #2)

Part 3: Calculating each symbol's bits and summarizing them up to get the total number of bits in the encoded article. (See attachment #1)

Part 4: Calculating the entropy of the article. (See attachment #2)

$$H = \sum_{j=1}^{m} P_j I_j = \sum_{j=1}^{m} P_j \log_2\left(\frac{1}{P_j}\right) \text{bits}$$

Results

After calculating and debugging, the entropy and total number of bits of the article are listed below.

Entropy	4.5153
Average bits	4.5435
Total bits	18533

In Matlab, we also get the average bits per message which is very close to the entropy.

Attachment #1

Symbol	Frequency	Probability	Huffman encoding	Bits	Total bits
0	3	0.00073547	00000100010	11	33
1	2	0.00049032	01101001011	11	22
2	1	0.00024516	011010010001	12	12
3	1	0.00024516	011010010000	12	12
4	1	0.00024516	011010010011	12	12
7	1	0.00024516	011010010010	12	12
9	1	0.00024516	0000010001101	13	13
	649	0.15910762	001	3	1947
-	17	0.00416769	01100111	8	136
,	40	0.00980633	0000111	7	280
;	7	0.00171611	100010110	9	63
:	8	0.00196127	011010001	9	72
!	3	0.00073547	00000100101	11	33
?	3	0.00073547	00000100100	11	33
.	30	0.00735474	1000100	7	210
‘	13	0.00318706	11001101	8	104
"	30	0.00735474	0110101	7	210
(	6	0.00147095	0000010000	10	60
)	6	0.00147095	110011101	9	54
a	281	0.06888943	0100	4	1124
A	7	0.00171611	100010101	9	63
b	53	0.01299338	110010	6	318
B	7	0.00171611	100010100	9	63
c	107	0.02623192	11000	5	535
C	2	0.00049032	01101001010	11	22
d	116	0.02843834	10000	5	580
D	3	0.00073547	00000100111	11	33
e	401	0.09830841	111	3	1203
E	1	0.00024516	0000010001100	13	13
f	43	0.0105418	0000110	7	301
F	3	0.00073547	00000100110	11	33
g	78	0.01912233	000100	6	468
G	10	0.00245158	011001000	9	90
h	152	0.03726404	00011	5	760
H	5	0.00122579	0110010010	10	50
i	232	0.05687669	0111	4	928
I	5	0.00122579	0000010111	10	50
j	3	0.00073547	00000101001	11	33
J	6	0.00147095	110011100	9	54
k	17	0.00416769	01100110	8	136
K	4	0.00098063	0110010011	10	40
l	100	0.02451581	000000	6	600
L	6	0.00147095	110011111	9	54
m	76	0.01863202	000101	6	456
M	6	0.00147095	110011110	9	54
n	229	0.05614121	1001	4	916
N	5	0.00122579	0000010110	10	50

o	218	0.05344447	1011	4	872
O	1	0.00024516	000001000111	12	12
p	58	0.01421917	011011	6	348
P	3	0.00073547	00000101000	11	33
q	2	0.00049032	10001011101	11	22
r	193	0.04731552	1101	4	772
R	3	0.00073547	1000101111	10	30
s	228	0.05589605	1010	4	912
S	7	0.00171611	011010011	9	63
t	277	0.0679088	0101	4	1108
T	19	0.004658	01100101	8	152
u	87	0.02132876	000010	6	522
v	44	0.01078696	0000011	7	308
V	2	0.00049032	10001011100	11	22
w	75	0.01838686	011000	6	450
W	5	0.00122579	0000010101	10	50
x	8	0.00196127	011010000	9	72
y	56	0.01372886	100011	6	336
z	13	0.00318706	11001100	8	104
Total	4079	1			**18533**

Attachment #2

```
Clear ;

symbols = [1:66]; % Distinct symbols that data source can produce

prob = [%See attachment #1 column 3]; % Probability distribution of symbols

[dict,avglen] = huffmandict(symbols,p); % Create dictionary

entropy = prob *log2(prob.^(-1));%Get the entropy of this article
```