Jesse Lingeman, Vijay Pemmaraju
lingeman@cs.umass.edu, vpemmara@cs.umass.edu

0. Contributions

Contributions were split fairly evenly: Jesse did the Huffman Code implementation, helped debug the Hamming Code implementation, and did part of the writeup. Vijay implemented the Hamming Code, helped with debugging overall, and did part of the writeup.

1. Table of Character Frequencies

We stripped out all punctuation and only alphanumeric characters and whitespace characters were left in.

Character	Frequency
\r	7222
\n	7222
space	58759
1	81
0	28
3	18
2	30
5	20
4	20
7	16
6	13
9	10
8	30
A	415
C	271
B	480
E	201
D	251
G	185
F	147

Character	Frequency
I	1970
H	850
K	13
J	36
M	361
L	210
O	183
N	225
Q	9
P	163
S	666
R	124
U	51
T	760
W	540
V	34
Y	234
X	2
a	20284
c	6239
b	3363
e	31607
d	10732
g	4575
f	5396
i	16209
h	16403

Character	Frequency
k	2054
j	209
m	6630
l	9979
o	20482
n	17185
q	193
p	4056
s	15409
r	15417
u	7936
t	23380
w	6188
v	2688
y	5336
x	401
z	123

3. Compression Factor

We achieve a compression factor of 1.8 (length of ascii bitstring / length of huffman bitstring) for our character set.

5. Hamming Expectations

A

We have set $\alpha = 0.02$ so we expect 2% of the bits to be erroneous.

B

We expect

C

D

6. Hamming Measurements on ASCII

A

There were 334324 characters in the document.

B

The binary length of the document was 2674592 bits.

C

The channel introduced 72683 total errors.

D

179575 codewords had 0 errors.

E

54989 codewords had 1 error.

F

7844 codewords had 2 errors.

G

Only 736 codewords had more than 2 errors.

H

There were 19385 bit errors in the decoded string.

I

Error correction failed for 8580 codewords.

J

This results in 12732 incorrect characters.

7. Huffman results

We found that after decompressing the Huffman coding there were 310541 character errors (92% error)! This happened because the Huffman code is greedily decoded and prefix coded, so if a bit gets flipped early on and is not recovered by error correction, then the rest of the document may be off. Even a single uncorrected bit flip can completely destroy the decompression of a Huffman code.