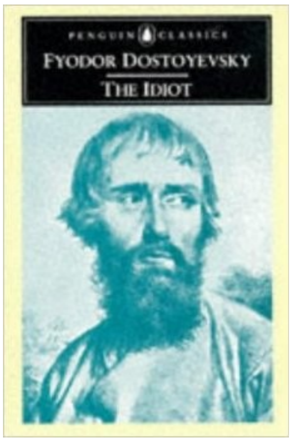
Telecom Network Systems

Fyodor Dostoyevsky

P. Stallinga

MIEET 4º ano

This analysis is based on the book of Fyodor Dostoyevsky, “The Idiot”



PENGUIN CLASSICS

FYODOR DOSTOYEVSKY

THE IDIOT

“Towards the end of November, during a thaw, at nine o'clock one morning, a train on the Warsaw and Petersburg railway was approaching the latter city at full speed. The morning was so damp and misty that it was only with great difficulty that the day succeeded in breaking; and it was impossible to distinguish anything more than a few yards away from the carriage windows.”
(Excerpt. Full text available based on Project Gutenberg)

In the analysis, convert to upper case and exclude punctuation marks: , . ? ! ( ) etc.

1) Using a simple code (a unique bit pattern for each character, without looking at the frequencies of the letters), how many bits would code the book?

2) Determine the frequency of letters in the full book
Based on this, how much information is there in the book (Shannon Entropy)?

3) Determine the first order Markov chain (which letter follows which letter).
Based on this, what is the amount of information in the book?

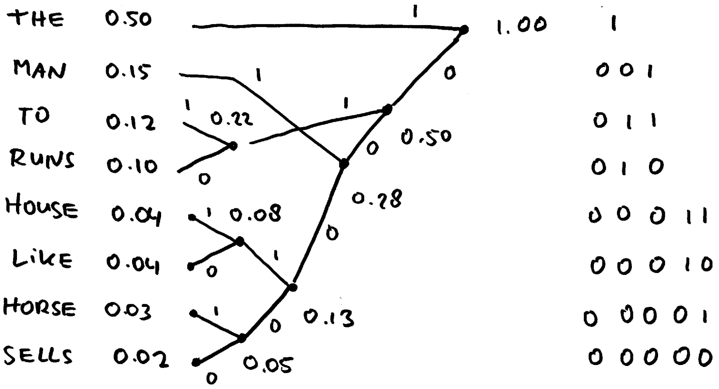
4) Determine the Huffman code based on the frequencies found in 2. How many bits would the book have in this coding scheme?

Huffman code (p. 94 of J.R. Pierce, “An introduction to information theory symbols, signals and noise”):

Suppose we want to encode a text containing the eight words *the*, *man*, *to*, *runs*, *house*, *likes*, *horse*, *sells*. In a frequency given in the table below:

Word	*Probability*
the	0.50
man	0.15
to	0.12
runs	0.10
house	0.04
likes	0.04
horse	0.03
sells	0.02

Without Huffman coding, without looking at the frequencies, we would need 3 bits per word, namely Log(8).



Huffman code works as follows: We first find the two lowest probabilities, 0.02 (*sells*) and 0.03 (*horse*), and draw lines to the point marked 0.05, the probability of either *horse* or *sells*. We then disregard the individual probabilities connected by the lines and look for the two lowest probabilities, which are 0.04 (*like*) and 0.04 (*house*). We draw lines to the right to the point marked 0.08, which is the sum of 0.04 and 0.04. The two lowest remaining probabilities are now 0.05 and 0.08, so we draw a line to the right connecting them, to give a point marked 0.13. We proceed thus until paths run from each word to a common point to the right, the point marked 1.00. We then label each upper path going to the left from a point 1 and each lower path 0. The code for a given word is then the sequence of digits encountered going left from the common point 1.00 to the word in question. The codes are listed in the table below:

Word	*Probability*	*Code*	*Number of Digits*	*Np*
the	0.50	1	1	0.50
man	0.15	011	3	0.45
to	0.12	001	3	0.36
runs	0.10	000	3	0.30
house	0.04	01011	5	0.20
likes	0.04	01010	5	0.20
horse	0.03	01001	5	0.15
sells	0.02	01000	5	0.10
				2.26

And we need only 2.26 bits per word!