Information Theory

Lab 6: Lossless compression – Huffman coding

Description of the tasks

All tasks realized during classes are `.pdf` files formatted similarly as this document. The tasks will be of different kinds. Every task will be appropriately marked:

- Tasks to be realized during classes are marked with □ – you won't get points for them, but you still need to do them.
- Pointed tasks to be realized during classes are marked with ◇ – you need to do them during class and show to your teacher, and in the case you don't manage to do so (or are absent) they become your homework (⋆).
- Homeworks are marked with ⋆ – they also have assigned a number of points, and you need to deliver them to your teacher before a deadline (usually before the next class).
- You may use any programming language you like for the programming tasks, but you are limited to only the standard library of that language and libraries widely accepted as standard.

Objective

During this class we will learn about Huffman coding, an optimal way to binary encode a set of symbols depending on their probability.

Preparation

- For this task will be needed text corpuses, which can be downloaded from http://www.cs.put.poznan.pl/ibladek/students/timkod/lab_huffman.zip.
- Texts are normalized and contain only 26 small letters of Latin alphabet, digits, and spaces (37 characters total).

1 Huffman coding *10pt*◇

Task

Implement Huffman coding algorithm using a priority queue. Implementation should contain the following functions (which will be also used in the next homework):

- **create** – creates a code (e.g., as a dictionary from a character to a binary sequence representing it) given a list of frequency of characters.
- **encode** – creates encoded representation of the text.
- **decode** – decodes encoded text.

Print the binary codes assigned to characters, and use them to compress the text corpus. Print the number of bits of the compressed text, and compare it with the number of bits of a shortest possible fixed-length encoding.

Trivia

Huffman coding was invented in 1951 by David Huffman during his studies on MIT. To pass information theory classes by prof. Robert Fano from, he could either write an exam or a written assignment. He chose the latter, and the topic was about effective encoding of information in binary code. He was close to giving up and starting learning for an exam, but then he got the idea of using a binary tree sorted over the frequency of appearance. He also proved the efficiency of this coding.

Huffman coding was better than the coding which Fano and Shannon were working at the time, known today as Shannon-Fano coding.

Sources

- https://en.wikipedia.org/wiki/Shannon%27s_source_coding_theorem
- https://en.wikipedia.org/wiki/Shannon%E2%80%93Fano_coding
- https://en.wikipedia.org/wiki/Huffman_coding
- http://math.harvard.edu/~ctm/home/text/others/shannon/entropy/entropy.pdf
- http://www.inference.org.uk/itprnn/book.pdf