Chair for Algorithms
and Data Structures
Prof. Dr. Hannah Bast
Claudius Korzen

Information Retrieval
WS 2017/2018

http://ad-wiki.informatik.uni-freiburg.de/teaching

UNI FREIBURG

Exercise Sheet 4

Submit until Tuesday, November 21 at **12:00pm (noon)**

Exercise 1 (6 points)

This exercise is about proving one direction of Shannon's famous source coding theorem.

In the lecture, we proved that for any prefix-free code it always holds that $\sum_i 2^{-L_i} \leq 1$, where L_i is the length of the encoding for integer i. Use this to show that for a random variable X drawn from $\{1, \ldots, m\}$, the expected code length $E(L_X)$ is always at least the entropy $H(X)$.

Hint: this is best done with Lagrangian optimization, as explained in Lecture 3.

Exercise 2 (8 points)

This exercise is about the optimal encoding for the gaps of the lists from an inverted index.

In the lecture we have shown that it is reasonable to assume that a fixed gap of an inverted list is distributed like a random variable X with X with $\Pr(X = i) = p_i = (1-p)^{i-1} \cdot p$ for some $p < 1$. Show that under this assumption, Golomb encoding with modulus $M = \lceil 1/p \cdot \ln 2 \rceil$ is an entropy-optimal encoding for the gaps, that is, $L_i \leq \log_2(1/p_i) + O(1)$ for all i.

Optionally, if you feel up to it, try to prove the stronger $L_i \leq \log_2(1/p_i) + 1$ for all i, or at least try to make the additive constant as small as possible. This is not necessary to get full points for this exercise.

Hint: you can use without proof that $1 + x \leq e^x$ for all real numbers x.

[turn over for more bits]

Exercise 3 (6 points)

This exercise is about calculating the space usage of an optimally gap-encoded inverted index (doc ids only, no scores) for a document collection with a total of N words from a vocabulary of m words.

Let us make the following assumptions. Let $L_1, \ldots, L_m$ denote the m inverted lists in order of descending lengths (longest list first). For simplicity, we assume that each document contains each word at most once, so that $\sum_{j=1}^{m} |L_j| = N$. We further assume that the list lengths are Zipf-distributed, that is, $|L_j|$ is proportional to $1/j$. We also assume that the gaps in each inverted list are randomly distributed, as explained in the lecture and assumed in Exercise 2, and that we have an optimal code for each L_j. (Golomb with the right modulus would be such an optimal code, but that is not important for this exercise.) Under these assumptions, the expected code length for a gap from L_j is $\log_2 j + O(1)$ bits. You can use this without proof.

Show that under these assumptions the expected total number of bits required to gap-encode all the inverted lists is $N \cdot (\log_2 m)/2 + O(N)$. That is, the average number of bits per posting is $(\log_2 m)/2 + O(1)$.

Hint: you can use without proof that $\sum_{j=1}^{m} 1/j = \ln m + O(1)$ and that $\sum_{j=1}^{m} (\ln j)/j = (\ln^2 m)/2 + O(1)$.

Commit your solutions in a single PDF in a new sub-directory *sheet-04* of your folder in the course SVN, and commit it. We recommend that you typeset your solution using LaTeX. With nice handwriting (if you are not sure if your handwriting is nice, it is not), you may also hand in a scan as a single PDF (no other file format will be accepted). In that case, take care that the scan has sufficient resolution and that the file is not too large (< 1 MB). Also commit the usual *experiences.txt*.

Of all the movies you have watched (or rewatched) this year, which one is your favorite and why?