Information Retrieval and Data Mining (IRDM) SS 2015
Prof. Dr.-Ing. Sebastian Michel
MSc. Koninika Pal
TU Kaiserslautern, FB Informatik – Lehrgebiet Informationssysteme
Sheet 5: Handout 17.06.2015, Presentation 30.06.2015

TECHNISCHE UNIVERSITÄT KAISERSLAUTERN

http://dbis.informatik.uni-kl.de

Assignment 1: Inverted Indexes (1 P.)

Consider an English document collection consisting of $|D| = 10^9$ documents, $|V| = 10^6$ terms, and a total of 10^{11} term occurrences.

(a) Estimate the number of bytes needed to represent the posting list for the most frequent term when using 64 bit document identifiers and 16 bit term frequencies? Here, you may assume that Zipf's law extends to document frequencies, i.e., the most frequent term occurs in 10 % of the documents. Further, for simplicity, you may assume that the term occurs with the same term frequency in all of these documents.

(b) Under the same assumptions, estimate the number of bytes needed to represent the posting list for the most frequent term when using variable-byte encoding for document identifiers (use gap encoding for sequence of documents) and Gamma encoding for term frequencies.

(c) Assume that we want to index bigrams in addition to single words. Can you provide an upper bound for the number of indexed terms (i.e., unigrams and bigrams) in our new dictionary? Can you provide an upper bound for the total number of postings in our new inverted index?

Assignment 2: Top-K Query Processing (1 P.)

Consider a top-k query with $m = 3$ terms, the user is interested in $k = 2$ results, and (non-weighted) summation as score aggregation. The underlying three index lists have the following (document identifier, score) entries:

L_1
d_1 0.9
d_7 0.7
d_3 0.3
d_2 0.3
d_4 0.3
d_5 0.2
d_6 0.1

L_2
d_3 0.8
d_4 0.8
d_7 0.5
d_1 0.3
d_6 0.2
d_5 0.2
d_2 0.2

L_3
d_1 0.7
d_6 0.6
d_7 0.5
d_4 0.4
d_2 0.3
d_3 0.2
d_5 0.1

(a) Apply the TA method (with random accesses) to this setting. Document all index accessing steps and the top-k after each of them. How many sorted accesses (SA) and random accesses (RA) does the method need?

(b) Apply the NRA method (no random accesses) to this setting. Document all index accessing steps, the top-k, and the set of candidates after each of them. How many sorted accesses does the method need?

Assignment 3: TAAT and NRA (1 P.)

(a) The TAAT method presented in the lecture processes conjunctive queries by reading posting lists one-at-a-time in ascending order of their corresponding term's document frequency. As TAAT proceeds, the set of accumulators shrinks, and the method terminates early once no accumulators are left. Assuming that you know for each word pair (u, v) the number of documents $df(u, v)$ that contain both words. How can you use this information to let TAAT terminate faster for queries with no results and/or use less main memory?

(b) The NRA method presented in the lecture is guaranteed to find the correct set of top-k results. However, it does not guarantee that the order of documents within the top-k is correct. How would you modify NRA algorithm, so that it returns the top-k results also in correct order? This should require adding no more than few lines of pseudo-code to NRA.