National University of Singapore
School of Computing
CS3245: Information Retrieval
Tutorial 2

Boolean Retrieval and the Term Vocabulary and Postings List

Readings: IIR, Chapters 1 & 2

1. Give the best strategy for calculating two following queries:

   (a) `(Singapore AND University AND National) OR NUS AND NOT Nanyang AND NOT Management`?

   (b) `(Natural AND Language AND Processing) OR (NLP AND NOT Neurolinguistic AND NOT Programming AND NOT Neuro-Linguistic)`?

   For both queries, obtain frequency counts for the individual terms from your favorite search engine. Treat the counts as the postings size for a Boolean search. Use a search for "the" to estimate the size of all documents in the search engine's index (which you will need for processing `NOT`. Note: you may need to make some assumptions about how to calculate the resulting size of conjunctive queries).

2. In lecture, we introduced the notion of **skip pointers**, to save on the number of comparisons needed to create the intersection listing for `AND` queries.

   (a) Skip pointers work for `AND` queries. Do they work for queries that exclusively have `OR` or `NOT`?

   (b) Does the use of skip pointers help in reducing the computational complexity of the intersection operation from $O(n + m)$?

   (c) Describe a configuration of two query terms n and m in which we would expect skip pointers to be utilized. Also describe the converse: describe a configuration of two query terms n and m in which we would be certain no skip pointers would be used.

3. **The Positional Index**

   (a) The standard positional index's structure is shown in Figure 2.11 of IIR, partially reproduced below, just showing the entry for "to" with 993,427 documents that match:

   to, 993427:
   (1,6: (7,18,33,72,86,231);
   2,5: (1,17,74,222,255);
   4,5: (8,16,190,429,433);
   5,2: (363,367);
   7,3: (13,22,191); ...)

   What if we flattened the positional list and stored position and document ID as a two-tuple (*e.g.*, $< docId, position >$), throwing away the frequency within the document. Describe how the above postings list for "to" would look like. Would this format

be preferable to the standard positional index described above? Describe the benefits/weaknesses of these two data structure formats, assuming that every number takes the same number of bytes to encode.

(b) Would skip lists be useful inside the positional lists' encoding? Why or why not?

4. **Token normalization.**

(a) We've covered stemming, case folding and eliminating stopwords as ways of shrinking the postings file and the dictionary. Rank these three methodologies in terms of how much space they may save with respect to the entire index for English. Justify your answers.

(b) The text describes problems with some queries when they undergo normalization through stemming or lemmatization. Describe at least two "academic" queries that would have such problems. Do a search on your favorite search engine and on LINC and see whether the problem that you describe manifests or not.

(c) Two companies have merged and their intranet search engines need to be merged together to form a single search engine. You have the dictionary and postings from both engines at your disposal. Can you simply merge the two dictionaries and postings files together? Why or why not? Describe in full.