National University of Singapore
School of Computing
CS3245: Information Retrieval
Tutorial 3

Tolerant Retrieval and Indexing

Readings: IIR Chapters 3, 4 & 5

1. **Wildcard queries and Permuterms.**

   (a) In English, many prefixes of word stems change the semantics of words ("un-", "anti-", "super-"), while suffixes often are used to inflect a word, or change its part of speech ("-s", "-ing", "-est", "-ly"). Given this information, do you think it more important to allow wildcards in the front or the back of word stems?

   (b) The textbook describes wildcards as interacting with stemming. If stemming is done by a search engine, what is the value of supporting wildcard queries?

   (c) `m*n` is a wildcard query that matches terms like "man", "munchkin" and "moon". If we were to use a permuterm index to find such matches, show which rotations of these three terms would match the query `m*n`.

   (d) How many entries would there be in the permuterm index for a word of length k?

   (e) From Table 4.2, RCV1 has 400,000 terms, where each term has 7.5 bytes (or characters per term). Given that the "standard" RCV dictionary index is 11.2 MB in size from Table 5.2, estimate the size of the dictionary if it were support permuterm queries.

2. **Spelling Correction.**

   (a) The book suggests that we use biword statistics to implement spelling correction. Your friend, Kno C. Guy, says that since storing biwords are expensive, that we should devise a context sensitive spelling correction system only using uniword (single word) statistics. Do you agree? Why or why not?

   (b) Second language learners of English (e.g., Chinese nationals learning English) often get similar words confused. Is it "the effect" or "the affect"? Or "the school principle" or "the school principal"?

   Let us say that you are given the search engine logs for a day for a large search engine, in which millions of user queries are given. Describe an algorithm to correct such word usage mistakes. How could you make the system more accurate but suggest less corrections? Or suggest more corrections at the cost of accuracy?

3. **Index Construction.**

   (a) Block sort-based indexing (BSBI) is the first algorithm introduced by IIR, and Single Pass In-Memory Indexing (SPIMI) the second. The book emphasizes that BSBI scales and works well but requires the terms to be uniquely mapped to Term IDs. Explain why.

   (b) In UNIX, processes have a limit on the number of file descriptors (file cursors) that can be in use at once (for example on sunfire, this limit is set to 256). This could limit both BSBI and SPIMI in the MergeBlocks step. Suggest a fix for this problem when a large index (say, with well over 256 blocks) needs to be built.

4. **Corpus Statistics.**

   (a) Zipf's law states that the frequency rank of a word is inversely proportional to its collection frequency (or vice versa). Imagine the word "the" is the most frequent word (rank 1) in a corpus with frequency 1,000,000. Estimate the number of *hapax legomena* (words that just appear once) in the corpus.

5. **Index Compression.**

   (a) Imagine we have a set of documents that just contain English alphabet lowercase letters as tokens (i.e., "a" – "z"; a document might be "a b c d e") and that we use these letters as dictionary terms without modification. Describe whether it is a good idea to use the "dictionary as a string" technique for compression in this scenario.

   (b) Assume we adopt the "dictionary as a string" technique in the above example. Also assume that the storage for term frequency, and term and postings pointer sizes are all 4 bytes. Show the percentage change in term lookup costs versus the savings in index size over an original, unblocked index, and for blocked indices where $k = 2$ and 4.