School of Computing and Information Systems
The University of Melbourne
COMP90049
Knowledge Technologies (Semester 1, 2018)
Workshop exercises: Week 7

1. What are the four primary components of a **Web–scale Information Retrieval engine**? Briefly describe our goal in each of them.

   - **Crawling**: finding and downloading as many documents as we can from the web (hopefully all of them, although this isn't possible in practice)
   - **Parsing**: turning each document into a list of tokens (or terms), probably by removing page metadata, case folding, stemming, etc.
   - **Indexing**: building an inverted index out of all of the tokens in our downloaded document collection. (We stop worrying about the original documents at this point.)
   - **Querying**: after the previous three steps have been completed (off-line), we are ready to accept user queries (on-line), in the form of keywords, that we tokenise (in a similar manner to our document collection) and then apply our querying model (e.g. TF-IDF) based on the information in the inverted index, to come up with a document ranking
   - (Optionally) **Additional things**: change the above ranking, based on ad-hoc application of certain factors, for example, PageRank, HITS, click-through data, zones, anchor text, etc.

2. Recall the (hypothetical) method of **crawling** given in the lectures:

   (a) Would this method be *effective* at solving the problem of crawling? Why or why not?

   - Somewhat, although it depends on having a large, random set of seeds, which isn't really possible in practice. The method will miss large numbers of pages that aren't linked to from pages in the "core" of the World Wide Web, as well as all sorts of rich data encoded in databases, etc.

   (b) Would this method be *efficient* at solving the problem of crawling? Why or why not?

   - An efficient approach depends on having a good model of **web page duplication**, so that we can decide (quickly) whether a given page has already been crawled. For example, having a hash function with respect to the page's contents (although consider how large the set is, and how difficult it would be to avoid collisions!).

3. **Canonicalisation** (of text) typically comprises **tokenisation** and **normalisation**. What are these generally accepted as referring to?
   (Note the terminology is not used consistently in the literature; for example "tokenisation" occasionally refers to all three ideas.)

   - Canonicalisation, here, means having a single representation of a text which we can use to sensibly compare one text with another (in our case, a document on the Web, and a query).
   - Tokenisation, here, means decomposing the larger document into smaller information–bearing units ("tokens") than we can compare against (the keywords present in) our query.
   - Normalisation, here, means transforming a token into a form which is generally representative of other instances of the same idea (for example, by correcting spelling).

   (a) What are some issues that arise when canonicalising text written in English?

   - From the lectures:
     - Stopwords
     - Stemming
     - Date/number formatting

  - Dialect variation
  - Spelling errors
  - Hyphenated tokens
  - Compound words
  - The genitive *'s*
- Note that English is **easy** compared to many languages!

(b) (EXTENSION) What are some issues that might arise when canonicaling text written in other languages?