CAB431 Week 3 Review Questions

Search Engine Architecture

A software architecture generally consists of software components, the interfaces provided by those components, and the relationships between them. An architecture is used to describe a system at a particular level of abstraction. Search engine components support two major functions, which we call the indexing process (the major components are text acquisition, text transformation, and index creation) and the query process (the major components are user interaction, ranking, and evaluation).

Question 1. (**Indexing Process**) Which of the following statements is false? and justify your answer.

(1) The text transformation component transforms documents into index terms or features.
(2) Index terms (e.g., words, sometimes simply referred to as "terms") are the parts of a document that are stored in the index and used in searching.
(3) A "feature" is more often used in the field of machine learning to refer to a part of a text document that is used to represent its content, which also describes an index term.
(4) Examples of other types of index terms or features are phrases, names of people, dates, and links in a web page.
(5) The set of all the terms that are indexed for a document collection is called the index distribution.

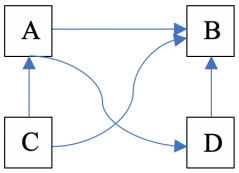
Question 2. (PageRank)

A Web graph $G = (P, L)$ consists of Web pages (vertices) and links (edges). The PageRank procedure takes a Web graph G as input and then outputs the better PageRank estimate PR using the following equation

$$PR(u) = \frac{\lambda}{N} + (1 - \lambda). \sum_{v \in B_u} \frac{PR(v)}{L_v}$$

where B_u is the set of pages that point to u, and L_v is the number of outgoing links from page v.

The following figure $G = (P, L)$ is a Web graph, where P = {A, B, C, D} and L = {(A, B), (A, D), (C, A), (D, B), (C, B)}.



Assume the initial estimate of each Web page is equally, i.e., $PR(A) = PR(B) = PR(C) = PR(D) = 0.25$; and $\lambda = 0.15$. Calculate the PageRank estimate (*PR* value) of each Web page after running the first iteration of procedure PageRank(*G*).

Web Crawler

To build a search engine that searches web pages, we first need a copy of the pages that we want to search. Unlike some of the other sources of text we will consider later, web pages are particularly easy to copy, since they are meant to be retrieved over the Internet by browsers. This instantly solves one of the major problems of getting information to search, which is how to get the data from the place it is stored to the search engine.

Finding and downloading web pages automatically is called crawling, and a program that downloads pages is called a web crawler.

Question 3. There are some unique challenges to crawling web pages. Identify which of the following is FALSE. You also need to justify your answer.

(1) The biggest problem is the sheer scale of the Web. There are at least tens of billions of pages on the Internet.
(2) Web pages are usually under the control of the people building the search engine database.
(3) The web crawler spends a lot of its time waiting for responses. It waits for the DNS server response, the connection to the web server to be acknowledged, and then the web page data to be sent from the server.
(4) Web pages are constantly being added, deleted, and modified. To keep an accurate view of the Web, a web crawler must continually revisit pages it has already crawled to see if they have changed in order to maintain the freshness of the document collection.

Question 4. (Removing Noise)

Many web pages contain text, links, and pictures that are not directly related to the main content of the page. Please identify which of the following statements is False and correct the false statement.

(1) A major component of the representation of a page used in a search engine is based on word counts; so, the presence of a large number of words unrelated to the main topic can be a problem.
(2) The simple technique based on the observation is that there are less HTML tags in the text of the main content of typical web pages than there is in the additional material.
(3) A document slope curve shows the cumulative distribution of tags in a web page as a function of the total number of tokens (words or other non-tag strings) in the page.
(4) The main text content of the page corresponds to the "plateau" in the middle of the distribution. This flat area is relatively small because of the large amount of formatting and presentation information in the HTML source for the page.
(5) The detection of the main content can then be viewed as an optimization problem where we find values of i and j to maximize both the number of non-tag tokens below i and above j and the number of tags between i and j.