## CS 345: Homework 2 (Due: Before midnight of Oct 18, 2016)

**Rules.** Individual homeworks; see Handout 1 (aka Syllabus).

**Problem 1.** (15 points) **Quality Assurance**

We perform the same query **CS 345 Web Search** in Bing and Google to find course related material. Results are on the course web-page, Section B with three files of links for Google and three for Bing with a total of 10 retrieved links each. A relevant result is considered one that relates to "CS 345 Web Search course at NJIT". (You might need to inspect a page closer.)

(a) Use the information available to determine and tabulate the results below determining which document is Relevant and rates of Recall and Precision after the introduction of each result (of the ten) for both search engines.

(b) Then go on establishing interpolated precision and effectiveness.

In computations, round up to the next integer (i.e. take a ceiling of a percentage point i.e. a 33.3% would become 34%).

```
HW2P1a:  GOOGLE                  BING            HW2P1b: GOOGLE                    BING
    Rlvnt|Recall Prcsion    Rlvnt|Recall Prcsion   Recall  Prec  InterP       Recall Prec InterP
d1 |     |      |       ||      |      |          0%     -                   0%     -
d2 |     |      |       ||      |      |       ------------------------------------------------
d3 |     |      |       ||      |      |         20%                       20%
d4 |     |      |       ||      |      |       ------------------------------------------------
d5 |     |      |       ||      |      |         40%                       40%
---|-----------------------------------------  ------------------------------------------------
d6 |     |      |       ||      |      |         60%                       60%
d7 |     |      |       ||      |      |       ------------------------------------------------
d8 |     |      |       ||      |      |         80%                       80%
d9 |     |      |       ||      |      |       ------------------------------------------------
d10|     |      |       ||      |      |        100%                      100%
---|-----------------------------------------  Effectiveness:             Effectiveness:             (6-point)
```

Figure 1: Tables for HW2 Problem (1a) and Problem (1b)

**Problem 2.** (18 points) **(Paper by Brin and Page)**

Read the paper (pdf and HTML through link L1 in section C5 of the course web-page). For Hashin consult wikipedia or your favorite CS114 or CS435 textbook. Justify your answers based on the info of the papers.
`http://en.wikipedia.org/wiki/Hash_table`
might also help.

- **(a)** According to the paper, what was the size (bytes) of Google's Lexicon around 1998?
- **(b)** According to the paper how many bits for a `docID`? Quote the paper.
- **(c)** Name/List the data structures used by Google for indexing only (not all of them are listed in the architecture figure).
- **(d)** In what data structure(s) does Google use binary search?
- **(e)** Does Google (1998) use a hash table for the dictionary? What do they use? Why ?
- **(f)** How many bytes are assigned to each hit (of the hitlist)? How many types of hits? What are they (types of hits)?

**Problem 3.** (10 POINTS) **(Heaps' Law)**
Is the state of Google's dictionary (lexicon) in 1998 (use paper L1 and Subject 2 for reference) consistent with Heaps' Law i.e. how many words does the lexicon maintain and how many unique words does Heaps' Law predict about the 150GB or so of the Corpus?

**Problem 4.** (12 POINTS) **Do the $w_1, w_2$ experiment**

Last time we tried this problem the best set of index-terms to estimate reliably the corpus size of Google and Bing was `pasta` and `physics` or `francaise` and `icosahedron`. The textbook's attempt of `tropical` and `Lincoln` does not work any more. And it is very likely that pasta and physics won't work either! We ask you to try find the three set of queries that do not include any of the previous index-terms nor anything similar to them (eg. mathematics instead of physics or polar instead of tropical) for $w_1$, $w_2$ and $w_1$ $w_2$ respectively so that the estimate for both engines is reasonable/reliable and over 20G but less than 50G. The same set will query Google and Bing. The $w_1$, $w_2$ and the reported hits will be included in E1, E2, E3 and the estimate of the Corpus size of Google and Bing will be shown in E4. Provide screenshots. Stick on using the first Google/Bing result page to capture the screenshot for the results.

| No | Query Term(s) | GOOGLE No of hits | BING No of hits |
|---|---|---|---|
| E1 | $w_1$ is $\langle$ $\rangle$ | | |
| E2 | $w_2$ is $\langle$ $\rangle$ | | |
| E3 | $w_1$ $w_2$ is $\langle$ $\rangle$ | | |
| E4 | Corpus size estimate is | | |

Table 1: Table for HW2 Problem 4

**Problem 5.** (14 POINTS) **Hash Table design circa 1998**
You have 2048MiB of main memory of which 548MiB are being used by the operating system plus related programs such as those manipulating tables $A$ and $T$ below. You are asked to organize the remaining space to support a hash table where words are stored in a contiguous table which is an array $A$ of characters delimited by a null character `\0`. A hash table $T$ will store a `wordID` along with a reference (pointer or index) $p$ to $A$. That way a `wordID` will be associated with a specific `word`. The average length of a `word` is given as 7 characters stored in UNICODE A `wordID` and $p$ can only be in multiples of two bytes (i.e. 2B, 4B thus 21bits or 24bits won't be an option) for efficiency. Organize $T$ and $A$ for maximum efficiency. In an efficient implementation answer the following questions by filling the data in the table that follows. Justify your answers and choices. Round to nearest million for $n, m$, $T, A$ but make sure you don't exceed the amount of available memory (i.e. $A+T$ should be accommodated easily by the available memory). Make sure that you fill the following table with data.

```
  n       =                    (a) How many words n can the scheme support,
  m       =                    (b) how big would the hash table size m (number of entries) be,
  wordID  =                    (c) how many bits for a wordID,
  p       =                    (d) how many bits for pointer p,
  T       =                    (e) how much space (bytes) will you scheme use for T,
  A       =                    (f) how much space (bytes) will you scheme use for A, and
A+T       =                    (g) what is the total space $A+T$ used by your scheme?
```

Date Edited: 9/23/2016