Programming Assignment 4

CS 410/510: Databases (Spring 2014)
100 points, Due Date 7th May (Wednesday), before 11:00 PM

Information Retrieval Using Unchanged SQL

- You should create a directory ~/cs410/lab4 for this assignment.

- Copy the files
  ~vijaydialani/cs410/lab4/files/prep.c
  and
  ~ vijaydialani /cs410/lab4/files/stops.txt
  to your assignment directory.

- Execute the following command (assuming you are in the directory ~/cs410/lab4) to make a link to the text file containing articles from the Wall Street Journal.
  Don't copy this file as it is quite large!
  ln -s ~vijaydialani/cs410/lab4/files/wsj7_001

- Compile the preprocessor with the following command:
  gcc -o prep prep.c

- Invoking the preprocessor without any arguments shows (as given below) what arguments it takes.
  onyx:prep
  Starting preprocessor: 0.5B
  Invalid Syntax: prep <infile> <outname> <start_doc>
  where: <infile> = name of input file that is a list of all input files.
  <outname> = filename for all output files
  <start_doc> = starting document number

- The preprocessor creates two files, one with an extension "*.doc*", which contains the doc table, and the second with an extension "*.index"*, which contains the doc term table.

Assignment

This assignment has two parts. The first part deals with creating the database and the appropriate tables. The second part concerns performing information retrieval using the database created in the first part.

1. Creating the database

Write a Java program (using JDBC API for MySQL) with the following options:

mktbls <database-name> <textfile>

This program performs the following actions:
(**item 1 and 2 have been implemented in the provided partial program**)

1. Runs the preprocessor and creates two files out.doc and out.index from a valid tagged file text file (You can assume that the text file is a valid file). For example if the file *infile* contains the name wsj7_001 then you will invoke the preprocessor as follows:

   prep infile out 0

   Note that your program creates the file infile.

2. Now convert the file out.doc to the file doc.unl using awk so that it is suitable for loading into the database using the load data command. Similarly convert the file out.index to the file doc term.unl and the file stops.txt to stop term.unl. After the conversion, delete the files out.doc and out.index.

3. Create the database with the user-supplied name and create the tables doc, doc term, stop term, idf and query. Load the tables doc, doc term and stop term using the load data command.

4. Now you need to figure out how to fill the table idf from the tables doc and doc term. Write a procedure that does this filling of the idf table.

Note: All of the above actions are to be carried out by your Java program. The database schema can be found in file ~vijaydialani/cs410/lab4/files/schema

Your program should at least report an error message or success when creating the database, selecting the database, creating the tables and filling of the tables. Make sure that your utility mktbls drops the old database if one exists before trying to create a new one. Also, display messages that keep the user informed as to what is going on.

---

2. Information retrieval using the database

Next you will need to finish a partial program ir.java, an information retrieval front-end utility, which uses the database created in the first part. The syntax of the command to invoke the program will be:

ir <database-name> <operation> [<k>] <query-file> <operation> <-- OR | AND | TAND | RELEVANCE

,where operation is either a Boolean OR, Boolean AND, Boolean TAND or RELEVANCE.

Your program should recognize any unique prefix of the operators and the case does not matter. In case of threshold AND there is an optional argument [<k>]. The file <query-file> contains the words of the query. The argument <database-name> should be a valid database created by the program mktbls as described in first part of the assignment. Note that for the operations OR, AND, and TAND, the result should be sorted by documents ids.

Queries

A reference set of queries is in directory ~vijaydialani/cs410/lab4/queries

Use the queries 151.query, 152.query, 153.query, 154.query, and 155.query. An example of the query contained in a file named 151.query is given below.

```
<top>

<num> Number:  151
<title> Topic:  Coping with overcrowded prisons
<desc> Description:
The document will provide information on jail and prison
overcrowding and how inmates are forced to cope with those
conditions; or it will reveal plans to relieve the overcrowded
condition.

<narr>  Narrative:
A relevant document will describe scenes of overcrowding that
have become all too common in jails and prisons around the
country.  The document will identify how inmates are forced to
cope with those overcrowded conditions, and/or what the
Correctional System is doing, or planning to do, to alleviate the
crowded condition.

</top>
```

Each query is contained in a separate file. The provided partial program ir.java parses the query into a “tmpfile/data structure” as follows.

- You can assume that each tag is on a separate line and starts that line.
- Take all the actual words in the title, description, and the narrative. That is, strip out the tags, the number line and other headings like Topic:, Description:, Narrative:

- While parsing the words, make sure to remove any leading or trailing punctuation as well as removing 's and s' at the end of a word.

After a query has been parsed, you need to write the function fill query table. The query table will contain all the words found in the query along with their frequency except the words found in the stop term table.

Submission

Submit your programs from onyx by copying the files (except "wsj7_001" and those query files) to a new empty directory (with no subdirectories) and typing the following FROM WITHIN this directory:

submit vijaydialani cs410 lab4