Project I

CS 410/510: Information Retrieval on the Internet
Due Tuesday April 20, 2010
10 points

Goal: The goal of this project is to create and use a simple search engine using Lucene. This will help you begin to understand how search engines work, and to gain some familiarity with Lucene that will help in future projects.

Details:
This project is to be done individually. You may discuss the project with your classmates, but the work you turn in should be your own. See the *Policies* and *Academic Integrity* sections in the course syllabus.

Lucene is an open source project from Apache that consists of a set of Java classes that can easily be assembled to create a search engine that can be modified and customized. Note that Lucene has been ported to other languages. You may use a language other than Java, but no support from the instructors will be available for any language other than Java. Therefore we highly recommend that all students do this and the other projects involving Lucene in Java.

The Java API is available online and can be very helpful: http://lucene.apache.org/java/docs/api/

How to do the project:

1. Go to http://www.apache.org/dyn/closer.cgi/lucene/java/ to download both Lucene 3.0 **and** the Lucene source code
2. Look at: http://lucene.apache.org/java/3_0_1/gettingstarted.html and the demo programs. When you downloaded Lucene you also downloaded the source code for the demos. (The path to the demo source should be something like:
   lucene-3.0.1\src\demo\org\apache\lucene\demo
   Use the Lucene API and the source code to help you understand how the commandline demo program works. The FAQs may be useful too.
   http://wiki.apache.org/jakarta-lucene/LuceneFAQ
3. Using IndexFiles.java and SearchFiles.java (in the demo) as a basis, create a simple commandline program. Feel free to copy as much of this code as you want but please name the directory that will contain your index ***yourname*_index** (e.g. "steinhauer_index" not just "index") so that I can run your program without overwriting another student's files. The files you will be indexing for this project are plain text. Look at the source code for StandardAnalyzer.java and any other classes that become necessary.
4. Download the zip file from: http://web.cecs.pdx.edu/~maier/cs510iri/Shakespeare/index.html Extract the text files and index them with your search engine. Do some simple searches to make sure it works.
5. Answer questions 1 – 6 below.
6. Now implement a custom stopword list that consists of (only) the following words:
   about, dost, from, hath, his, O, that, the, thou

7. Implement stemming in your search engine using the Porter stemming algorithm. [Hint: **don't** implement the algorithm yourself, let Lucene do the work. Look at the analysis package.] Remember that both the indexing program and the searching program have to use the same protocol for stemming and stopword removal.
8. Answer questions 7 – 10 below.

Questions:

You will need to use the Lucene API and/or Lucene source code for answering some of these questions.

1. Does the commandline demo system use stopwords? How do you know? If it uses stopwords, what stopwords does it use (list them)?
2. Does the commandline demo system do stemming? How do you know?
3. How does Lucene make searches case insensitive (in the demo)?
4. What does Lucene do with apostrophes (in the demo)? How does it do it? (Your answer should state what is done, by what method, of what class).
5. Submit the following queries to your search engine and report the hits by listing them in order. (Note: this is to be done before you do steps 6 and 7).
    a. about the prince
    b. his fiery sword
    c. *ALAS O Love*
    d. love
    e. loves
6. Why do you think the hit order is different between queries *c* and *d*? Speculation is fine; you don't have to figure out the Lucene scoring algorithm at this point.
7. Submit the following queries to your search engine and report the hits by listing them in order. (Note: this is to be done **after** you do steps 6 and 7 above).
    a. about the prince
    b. his fiery sword
    c. alas o love
    d. loves
8. Does it matter whether stemming occurs before or after stopword removal? (Consider this as a general question, not just with respect to these documents or stopwords). Why?
9. What does the Lucene *Field* class do? A one or two sentence summary is all that is necessary.
10. How might you (programmatically) get the frequency of a particular term in a particular document that has been indexed by Lucene? What method of what class could you use?

Deliverables:

1. Turn in your answers to questions 1 – 10 as hard copy at the beginning of class on April 20, 2010. Make sure your answers are legible. Word processed is best.
2. Submit the code for your search engine by emailing your .java files (not the Lucene files) to jsteinha@cs.pdx.edu by 10:00 am on April 20th.
   Be sure to put *cs510: Project I* in the subject line to make sure your homework is not overlooked.