UCI CS 221 Information Retrieval

Project 3 – Search Engine

Due dates: 2/17, 3/3 and 3/15

This assignment is to be done in groups of 2, preferably the same groups that were in place for Project 2 (but this is not necessary). Although this is presented as one single project that will take until the end of the quarter to complete, internally it is organized in 3 separate milestones, each with a specific deadline, deliverables and score. In doing milestones #1 and #2, make sure to look at the evaluation criteria not just of those milestones but also of milestone #3 –part of the milestones' evaluation will be delayed until the final meeting with the Reader.

You can use code that you or any classmate wrote for the *previous* projects. You cannot use code written for *this* project by non-group-member classmates. Use code found over the Internet at your own peril -- it may not do exactly what the assignment requests. If you do end up using code you find on the Internet, you must disclose the origin of the code. **As stated in the course policy document, concealing the origin of a piece of code is plagiarism**.

Use the Message Board for general questions whose answers can benefit you and everyone.

Goal: Implement a complete search engine for the ICS domain. At the end of this project, you should have a web interface that provides the user with a text box to enter queries and returns relevant results.

Implementation Choices

You have two implementation choices:

1) You can use Apache Lucene (http://lucene.apache.org/core/) or one of its other language bindings, or any other similar text search libraries
2) You can implement the indexing, querying and scoring by yourself

Your decision should take the following into consideration:

- **whatever you choose, you need to understand the details of how it works.** You need to be able to answer questions such as "What's the format of the index?", "What is the scoring formula?", etc.
- **if you use an external search library, the expectation regarding the overall quality of the project will be higher than if you implement everything by yourself**. I don't expect you to be able to match what Lucene does in 6 weeks; for projects implemented from scratch, lower quality is ok.

Milestones Overview

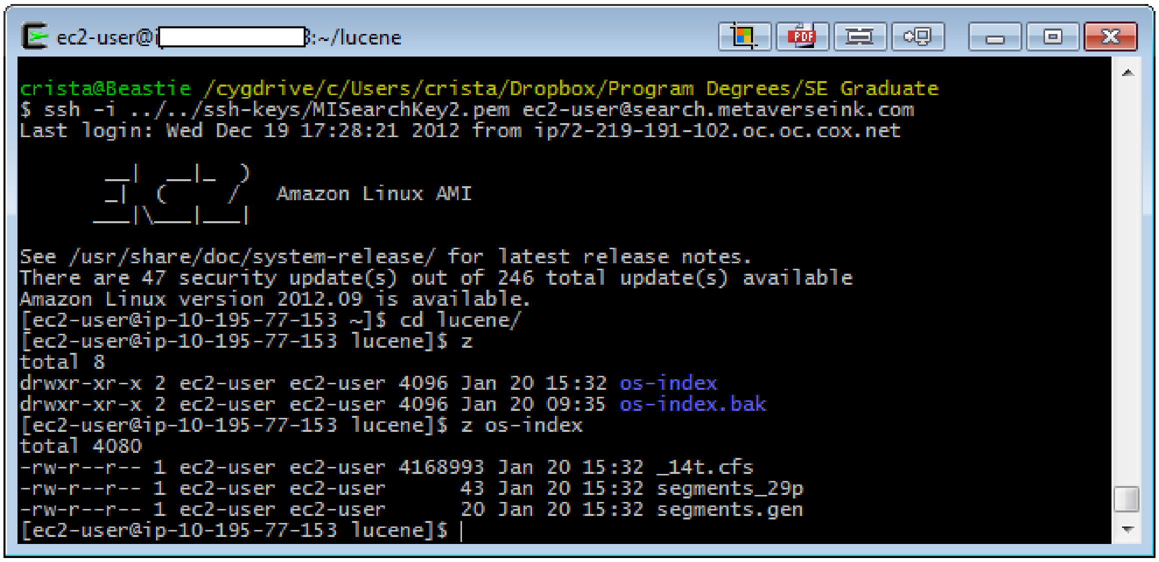
	Deadline	Goal	Deliverables	Contribution for score
#1	2/17	Produce an index for the ICS web pages	Short report	10%
#2	3/3	Search performance improvement	Short report and code	10%
#3	3/15	Produce the complete search engine, including UI	Demonstration	80%

Milestone #1

Using the pages that you stored by crawling the ics.uci.edu domain in the previous project, construct an index that maps words to documents (pages). The UI can be as simple as you need it to be in this phase. If you are using Lucene, make sure to take advantage of Luke (http://www.getopt.org/luke/).

Deliverable: submit a report (pdf) to EEE with the following content:

a) A picture of your index file(s) in your file system. Something like this:



```
crista@Beastie /cygdrive/c/Users/crista/Dropbox/Program Degrees/SE Graduate
$ ssh -i ../../ssh-keys/MISearchKey2.pem ec2-user@search.metaverseink.com
Last login: Wed Dec 19 17:28:21 2012 from ip72-219-191-102.oc.oc.cox.net

       __|  __|_  )
       _|  (     /   Amazon Linux AMI
      ___|\___|___|

See /usr/share/doc/system-release/ for latest release notes.
There are 47 security update(s) out of 246 total update(s) available
Amazon Linux version 2012.09 is available.
[ec2-user@ip-10-195-77-153 ~]$ cd lucene/
[ec2-user@ip-10-195-77-153 lucene]$ z
total 8
drwxr-xr-x 2 ec2-user ec2-user 4096 Jan 20 15:32 os-index
drwxr-xr-x 2 ec2-user ec2-user 4096 Jan 20 09:35 os-index.bak
[ec2-user@ip-10-195-77-153 lucene]$ z os-index
total 4080
-rw-r--r-- 1 ec2-user ec2-user 4168993 Jan 20 15:32 _14t.cfs
-rw-r--r-- 1 ec2-user ec2-user      43 Jan 20 15:32 segments_29p
-rw-r--r-- 1 ec2-user ec2-user      20 Jan 20 15:32 segments.gen
[ec2-user@ip-10-195-77-153 lucene]$
```

b) A table with assorted numbers pertaining to your index. It should have, at least the number of documents, and the number of [unique] words.

Evaluation criteria:

- Does the picture show a plausible index file?
- Are the reported numbers plausible?

Milestone #2

The following queries should be used to tune the performance of your search engine:

1 - mondego
2- machine learning
3- software engineering
4 - security
5 - student affairs
6 - graduate courses
7- Crista Lopes
8 - REST
9 - computer games
10 - information retrieval
(Feel free to use more)

The Oracle (i.e. ground truth) for the ranking of the results is Google. Write a program that sends these 10 queries to Google appended with "site:ics.uci.edu" and returns Google's search results. If your ICS crawler of project 2 stored only HTML pages, you should eliminate non-HTML documents from the Google results. For each query, the ordering of the Google results should be considered the "right order."

1) Compute NDCG@5 for your search engine before any performance improvements.
2) Improve your search engine's NDCG@5.

Deliverable: submit a zip file to EEE containing all the tools/scripts you wrote for this milestone as well as a report (pdf) with the following content:

a) Your NDCG@5 for 1) and 2)
b) Explain what you did in order to improve NDGC@5.
c) Include any other information that may be relevant for this milestone

Evaluation criteria:

- Is your second NDCG@5 good enough and/or were you able to improve it wrt the first?

- How general are the heuristics that you employed to improve the performance?

Milestone #3

Finish your search engine by giving it a web UI.

Deliverable: you will meet with the Reader and show him your search engine in action. The reader will ask you questions about any aspect of your implementation.

Evaluation criteria:

- Does your search engine work as expected of search engines?
- Are the search results for the test queries in milestone #2 good enough?
- How general were your methods/tools for milestone #2?
- How complete is the UI? (e.g. links to the actual pages, snippets, etc.)
- Do you demonstrate in-depth knowledge of how your search engine works? Are you able to answer detailed questions pertaining to any aspect of its implementation?