Course Project: Building Your Own Vertical Search Engine

Important Dates

* Project Topic Signup: Friday 18 September, 2020 at shorturl.at/iB456

* Proposal Presentation: Lecture 8 (Exact date TBA)

* Final Presentation: Last Lecture (Exact date TBA)

* Project/Document Submission Deadline: Wednesday December 9, 2020 11:55PM

Update Logs

08/09/2020 Initial Version

---

1. Goal & Important Note

This project is an open-ended exploration into the implementation aspect of information retrieval and web search. You will have to work in a team of 3 from the same section (Explicit permission from the instructors is required to form a group of fewer or more than 3 members), and design and implement a prototype search system that performs a task relevant to the broad field of IR. The project will be graded based on the novelty of your solution to the problem, the execution (data collection, system implementation and evaluation), and the presentation (written report and oral presentation). You may not use this project to satisfy other courses' requirements; however, after your project has finished, you may ***make significant extensions*** to your project to satisfy the requirements of other courses.

Note01: If you are not sure whether your project is related to IR or not, ask yourself these simple questions:

1. What are target user groups?
2. For each target user group, what are their information needs?
3. How can your system help to satisfy these information needs?
4. Can you fill in the blank? "We are implementing a search engine for _____."

Note02: You must implement a working prototype/system that demonstrates your idea. You must be able to demonstrate how your system works in the final presentation.

2. Scope and Ideas

You will be implementing a prototype vertical search engine for a particular data of your choice. Your search engine must contain at least the following components:

- Document Collection and Processing
- Document Indexing
- Document Ranking

Your search engine must handle unstructured data, and ***cannot simply use a database lookup (e.g. MySQL) to search***. You are encouraged to use open-source search system such as Elasticsearch[1], Apache Solr[2], Sphinx[3], etc. If you wish to use one of these open source search systems, you are responsible to learn how to use it by yourself.

3. Deliverables

3.1 Proposal Presentation

[1] https://github.com/elastic/elasticsearch
[2] https://lucene.apache.org/solr/
[3] http://sphinxsearch.com/

Prepare an 8-minute presentation that answers the following questions:

a.) What is your search system for?
b.) What are your target user groups?
c.) What is your data source?
d.) How to preprocess your data?
e.) How to index the documents?
f.) How to rank the documents?

3.2 Final Presentation

Prepare an 8-minute presentation that show the following:

a.) Demonstration of your search system

b.) Explain the technical difficulties, challenges, and lessons learned.

3.3 Project Submission

1. Write a project report (at least 2 pages) that elaborates the following items:
    a. Introduction
    b. Problem(s) that you are trying to solve
    c. Literature review or existing relevant systems
    d. Methodology
    e. Implementation
    f. Results and Discussion
    g. Conclusion
2. Create a new directory and name it `CourseProject_<id1>_<id2>_<id3>`, e.g. `CourseProject_6188111_6188222_6188333`. Let us call this the submission directory.
3. Put the project report, data, source codes, and other related files in the submission directory. If your project contains large files, you can put a link to download your solutions in the report.
4. Use compress the submission directory using either .zip or .7z format and produce the ***submission package,*** e.g. `CourseProject_5988111_5988222_5988333.7z`. Do not submit .rar files.
5. One of your team members then submits the submission package on MyCourses before the due date.

**Note on presentation:*

1. All presentations will be online. Presentation ordering will be announced after the project signup.
2. Since the presentations will be during class time, all students must attend the whole lecture where presentations take place and are encouraged to constructively provide feedback and ask questions.

4. GRADING

Students are expected to maintain standards of academic honesty and integrity. Violations of academic honesty and integrity in this assignment are unacceptable. Each group must work on their own project independently. A full or partial copy from other groups' solutions may automatically result in a zero score and a fail (F) grade for this course. Below is the tentative break-down of your project score:

Project proposal:	30%
Project final presentation:	40%
Project report:	30%

Need help with the project?

If you have questions about the project, please first post them on the forum on MyCourses, so that other students with similar questions can benefit from the discussions. If you still have questions or concerns, please come see one of the instructors during the office hours, or make appointments in advance. We do not debug your code via email. Consulting ideas among friends is encouraged; however, the code must be written by members in your own team without looking at other teams' code. (See next section.)

Academic Integrity

Don't get bored about these warnings yet. But please, please do your own work. Though students are allowed and encouraged to discuss ideas with others, the actual solutions must be written by themselves without being dictated or looking at others' code. Inter-team collaboration in writing solutions is not allowed, as it would be unfair to other teams. It is better to submit a broken program that is a result of your own effort than taking somebody else's work for your own credit! Students who know how to obtain the solutions are encouraged to help others by guiding them and teaching them the core material needed to complete the project, rather than giving away the solutions. **You cannot keep helping your friends forever, so you would do them a favor by allowing them to be better problem solvers and life-long learners. ** Writing code is like writing an essay. If each of you writes your own code, then there is almost no chance that your codes will appear similar. Your code will be compared with other students' and online sources using state-of-the-art source-code similarity detection algorithms. If you get caught cheating, serious actions will be taken!