INST 408Y Syllabus, Spring 2022

Douglas W. Oard (oard@umd.edu)

Preliminary Version as of November 19, 2021 (warning: some details may change!)

Background:

INST 408Y is a new elective course for the iSchool's B.S. in Information Systems (BSIS) major.

Course title: Search Engines

URL: http://users.umiacs.umd.edu/~oard/teaching/408y/spring22/

Learning Objectives:

Upon completion of this course, students will be able to:

- Articulate the capabilities and limitations of search engines.
- Describe design considerations when applying search engines to specific applications.
- Implement a search engine for a specific task.
- Compare the effectiveness of alternative search engine designs.

Modality:

This is a face-to-face course that meets from 9:30 to 10:45 AM on Mondays and Wednesday in ESJ 2309.

Approach:

The objective of the course is to give students a detailed understanding of current methods for creating systems that help people search for unstructured content, for which we will adopt text search as our driving example. The course is designed as 7 modules of 1.5 to 2.5 weeks each. The first four modules (indexing, traditional methods, evaluation, extensions) will help students build the basic understanding they will need to design, implement and evaluate their term project during the second half of the semester. The second half of the course then focuses on more advanced topics, starting with neural methods, then progressing to Web search, and finally illustrating how similar approaches can be extended to other applications (e.g., searching speech or foreign languages). A typical class session will include a mix of lecture and lab, with the lab session conducted using Jupyter notebooks. Reading assignments in one of two textbooks, and four programming assignments, will supplement the lectures. The course includes a term project in which students will select from a set of retrieval tasks for which suitable resources have been assembled in advance by the instructor; students may also design their own project, subject to instructor approval.

Textbooks:

- W. Bruce Croft, Donald Metzler and Trevor Strohman, Search Engines: Information Retrieval in Practice, 2015 Update. This book is available free on the Web. It will be the main textbook for the class, covering every aspect except the newly introduced neural models.
- Jimmy Lin, Rodrigo Nogueira and Andrew Yates, Pretrained Transformers for Text Ranking: BERT and Beyond, Synthesis Lectures on Human Language Technologies, Morgan & Claypool, to appear. Final preproduction preprint is available as free on the Web as arXiv:2010.06467 v3, August, 2021.

Schedule, Topics, Readings and Assignments

Session	Date	Module	Topic	Reading	Assignment
1	Monday, January 24, 2022		Introducing Information Retrieval		
2	Wednesday, January 26, 2022		Python refresher		
3	Monday, January 31, 2022	Indexing	Inverted indexing: Overview	Croft pp. 13-22	
4	Wednesday, February 2, 2022	Indexing	Tokenization and Stemming	Croft pp. 86-100	
5	Monday, February 7, 2022	Indexing	Building a simple inverted index	Croft pp. 125-133	
6	Wednesday, February 9, 2022	Indexing	Proximity indexing	Croft pp. 134-140	
7	Monday, February 14, 2022	Traditional Methods	Set retrieval using Boolean queries	Croft pp. 233-236	Inverted index
8	Wednesday, February 16, 2022	Traditional Methods	Ranked retrieval using BM25	Croft pp. 237-251	
9	Monday, February 21, 2022	Traditional Methods	pyserini		
10	Wednesday, February 23, 2022	Evaluation	Evaluation measures	Croft pp. 308-321	
11	Monday, February 28, 2022	Evaluation	Constructing test collections	Croft pp. 299-304	Evaluation
12	Wednesday, March 2, 2022	Evaluation	Significance testing	Croft pp. 325-338	
13	Monday, March 7, 2022	Improvements	Blind relevance feedback	Croft pp. 252-266	
14	Wednesday, March 9, 2022	Improvements	The sequential dependence model	Croft pp. 452-459	
15	Monday, March 14, 2022	Improvements	Learning to rank	Croft pp. 283-287	Parameter tuning
16	Wednesday, March 16, 2022		Project options		
	Spring Break				
	Spring Break				
17	Monday, March 28, 2022	Neural Methods	Neural Methods: Overview	Lin pp. 4-19	
18	Wednesday, March 30, 2022	Neural Methods	Huggingface BERT	Lin pp. 46-61	
19	Monday, April 4, 2022	Neural Methods	Pointwise reranking	Lin pp. 67-76	
20	Wednesday, April 6, 2022	Neural Methods	Pairwise reranking	Lin pp. 87-97	
21	Monday, April 11, 2022	Neural Methods	Result fusion	Croft pp. 438-442	Neural reranking
22	Wednesday, April 13, 2022	Web Search	Web search: Overview		
23	Monday, April 18, 2022	Web Search	Crawling	Croft pp. 31-45	
24	Wednesday, April 20, 2022	Web Search	Implicit feedback		
25	Monday, April 25, 2022	Web Search	Snippet generation	Croft pp. 215-225	Preliminary project report
26	Wednesday, April 27, 2022	Applications	CLIR	Croft pp. 226-232	
27	Monday, May 2, 2022	Applications	Multimodal retrieval	Croft pp. 470-479	
28	Wednesday, May 4, 2022	Applications	Question answering	Croft pp. 466-470	
29	Monday, May 9, 2022		Exam review		Term Project
Exam	May 14, 2022		Take-home final exam due		

Assignments

Students will complete five assignments, always due before class on a Monday. Students will work individually on the first four assignments and in two-person teams for the final project.

1. **Inverted index.** The goal of this assignment is to help students to understand the material taught in the indexing module. Students will write a python program to read a set of five short documents, tokenize those documents and stem the resulting tokens, build an inverted index, and use that index to list the documents in which each term appears.
2. **Evaluation.** The goal of this assignment is to help students to understand the material taught in the indexing module and in the evaluation measures session, and to prepare them for the discussion of the other two evaluation sessions. Students will use pyserini to index the TREC Robust track test collection and then will produce ranked lists for each title query using three ranking functions. They will then compute three standard evaluation measures for each system and for each query, and they will be asked to answer questions about why some systems do better for some queries by some measures.
3. **Parameter tuning.** The goal of this assignment is to give students practical exposure to the idea of parameter optimization, which will be useful background when discussing automated parameter optimization methods in learning to rank and neural methods. Students will be asked to use pyserini to sweep across possible parameters for the BM25 ranking function and for the number of documents and terms used for feedback in blind relevance feedback in order to select the best parameters on a training partition of the TREC Robust track test collection

queries, and then to compare their best parameter values with standard parameter values on a test partition of queries for that same test collection. They will be asked to answer questions about whether and why the improvements seen on the training partition do or don't match those seen on the test partition.

4. **Neural reranking.** The goal of this assignment is to demonstrate to students the substantial gains in retrieval effectiveness that are possible when using neural methods, and to give them experience with the computational costs involved. Students will use castorini (an extension to the pyserini system they used in the prior two assignments) to create a reranking cascade in which the first stage uses BM25 and the second stage uses a pointwise BERT reranker that has been pretrained on MS-MARCO.
5. **Term Project.** The goal of this assignment is to help students learn to apply ideas from the course to new tasks. Students will be assigned in two-person teams and each team will be assigned one of three information retrieval test collections for a task that has not (to that point) been specifically taught in the course and asked to build the best system that they can for that task. The three test collections will be from the TREC CaST track (retrieval in response to queries posed as part of a conversation between user and system), the CLEF ARQMath lab (finding answers to questions involving math by searching in earlier Math Stack Exchange questions and answers), or the TREC NeuCLIR track (finding Chinese documents using English queries). Teams will be assigned to test collections in a balanced way based on student preferences. In each case, a baseline system will be provided that the students can use as a starting point, and the goal will be to achieve statistically significant improvements over the baseline. A leaderboard will be provided that can be used by students to register their intermediate results. The winning team for each task will be exempted from (and will receive full credit for) the final exam.

Final Exam:
The final exam will be a two-hour open book, open notes, open-Internet written exam in which the students will answer questions about course concepts. No programming will be required.

Grading:
Class participation (for the lab session portion of each class): max of 20 points
First four assignments: max of 10 points each
Term project: max of 20 points
Final exam: max of 20 points
Final grades will be assigned with standard breakpoints (e.g., 90-92 is an A-), with no curve.

Prerequisite:
INST 326 (Object Oriented programming for Information Science) or equivalent. Rationale: Knowledge of python programming will be required.

Technology requirements:
In-class lab sessions and at-home programming assignments will be supported by code examples presented using Jupyter notebooks, so students will need a laptop and Internet access to use those cloud services both in class and from home. In the second half of the semester, students will use colab notebooks for cloud access to a GPU or TPU for the one homework involving neural methods, and (at their option) possibly also for their term project.