NYU Tandon School of Engineering
Computer Science and Engineering
CS 6913, Fall 2021

September 2, 2021

CS 6913: Web Search Engines

Course Prerequisites: Solid Computer Science background. Excellent programming skills, preferably in C/C++. General experience with the web and with HTML is expected. Knowledge of algorithms (equivalent to CS6033 or CS3414) is strongly recommended. Useful but not required are basic knowledge of networking and of Unix network and systems programming, scripting languages, and basic OS and database concepts. No background in machine learning or AI is required, though having such a background can be beneficial when choosing a course project.

Note on In-Person Attendance: This is an in-person course, with a few students attending remotely by special permission. Students are expected to attend class in person, and recordings may only be available to students who are remote. If you are an MS student taking the course remotely, you must have an accommodation approved by Tandon to do so. Every attempt will be made to accommodate remote students fairly, and to also accommodate students in different time zones. If you have any concerns that a particular measure or approach does not work for your situation, please let the instructor know promptly.
If you are in-person but expect to miss a particular class due to a valid reason, let the instructor know via email. Valid reasons include late arrival at the beginning of the semester due to visa or flight problems, or not feeling well or having to quarantine. You do not need to let the instructor know anything about your health status as this is confidential – instead, contact student services and they can let me know that you have a valid reason.

Time and Location: F 2:00–4:30pm, room RH605 in Rogers Hall (the main building on the Brooklyn campus).

Instructor: Prof. Torsten Suel, email: `torsten.suel@nyu.edu`. Office: Room 856, 8th floor of 370 Jay.

Office Hours: To be determined. Office hours will be mostly on Zoom, and there will be at least two different office hours, to accommodate students in different time zones.

Course Webpage: The course page will be at `http://engineering.nyu.edu/~suel/cs6913/`. We will also use Brightspace for important announcements and homework submissions, and may use Piazza or a similar system for course discussions.

Textbooks: The following book is recommended as background reading for this course. See also the course page for additional resources.
– Introduction to Information Retrieval, by Manning, Raghavan, and Schütze, Cambridge University Press, 2007 (available online for free).

Grading Policy: The final grade will depend on course assignments (35%), course project (30%), and a final exam (35%). The project also requires writing a longer paper describing the work and results.

Policy on Academic Dishonesty: Please see our policy on academic dishonesty on our schools website at `https://engineering.nyu.edu/campus-and-community/student-life/office-student-affairs/policies/student-code-conduct`. Common examples of misconduct include cheating, fabrication, plagiarism, and/or unauthorized collaboration. Students are expected to work on their own, with the possible exception of group projects or assignments if allowed by the Professor. Students may discuss work with other individuals either in the class or outside the class, but they *may not reuse code, results, or text received or retrieved from any source unless clearly disclosed in their submissions*. Any student who is found to be violating this policy will be given a failing grade for the work and will be reported, as described in the NYU Student Code.

Moses Center Statement on Disability: If you are a student with a disability who is requesting accommodations, please contact New York Universitys Moses Center for Student Accessibility (CSA) at 212-998-4980 or `mosescsd@nyu.edu`. You must be registered with CSA to receive accommodations. For more information, see `http://www.nyu.edu/students/communities-and-groups/students-with-disabilities.html`. The Moses Center is located at 726 Broadway.

Assignments: There will be several programming and written assignments. General discussions between students, and help in *setting up and using any programming environments*, are permitted and encouraged. However, no code or solutions may be copied! In the second half, students will work on a larger project, individually or in small groups. A list of available projects will be presented, and students may also propose relevant projects of their own interest.

Excused Absences: See the NYU Tandon policies and procedures on excused absences located at this link: https://engineering.nyu.edu/campus-and-community/student-life/office-student-affairs/policies#chapter-id-30199. In short, an absence can be excused if you have missed no more than 10 days of school. If an illness or special circumstance has caused you to miss more than two weeks of school, please refer to the section labeled Medical Leave of Absence. Students may request special accommodations for an absence to be excused in the following cases: (1) Medical reasons, (2) death in the immediate family, (3) personal qualified emergencies (documentation must be provided), and (4) religious expression or practice. Deanna Rayment, deanna.rayment@nyu.edu, is the Assistant Director of Student Advocacy, Compliance, and Student Affairs, and handles excused absences.

Assignments: There will be several programming and written assignments. General discussions between students, and help in setting up and using any programming environments, are permitted and encouraged. However, no code or solutions may be copied! In the second half, students will work on a larger project, individually or in small groups. A list of available projects will be presented, and students may also propose relevant projects of their own interest.

Course Outline: This course will cover a variety of topics related to web search technology. The main focus will be on large-scale web search engines (such as Google, Bing, or Baidu) and the underlying architectures and techniques. You will learn how search engines work, and get hands-on experience in building search engines from the ground up. You will also learn about the fundamental challenges and bottlenecks in current search technologies, and about research efforts that address these issues. In this context, we will also cover fundamentals of information retrieval, data compression, and computing with massive data sets. Here are some typical topics:

- basic web and search engine architecture
- web crawling and text indexing
- text and data compression
- I/O-efficient computing and computing with massive data sets
- search engine query execution and ranking functions
- computational advertising
- web mining, web graph mining, query log mining, and personalization
- retrieval evaluation techniques and measure
- relevance feedback, query expansion, and retrieval models
- complex rankers based on learning to rank and neural approaches
- advanced search engine architecture

The course will cover both algorithmic techniques and implementation aspects. Students will be required to complete several substantial programming projects in C/C++, Java, or the Python scripting language. Students will also have to read a number of research papers.

Policies and Suggestions. Following are a few tips on how to approach the assignments and projects.

(1) Start the assignments early, as they may be more substantial than they appear on first sight. Others may be easier, once you figure out the necessary parts. But they all need some planning, so a last-minute approach is unwise.

(2) Do not reinvent the wheel! For example, Python has a lot of useful functions in its standard modules. Check it out. You many find other useful libraries on the web. Of course, you should not use a libary that completely solves the entire assignment on its own – if in doubt, ask.

(3) Do not skip programming assignments! You should do all of them. Subsequent assignments may reuse code from earlier ones. Make sure you write your code such that you can modify it easily later.

(4) Discuss the assignments with classmates. Seek and provide help about things such as the programming environment, library functions, design decisions. Don't get stuck for hours because you cannot figure out, e.g., how to run a Python program or how to use a particular library - ask somebody. Ask other students, then ask the instructor if necessary.

(5) But do not cross the line. Do not copy code and solutions from others, and disclose any outside resources you have used as part of the readme file for the assignment.

(6) If you are in serious danger of falling behind, talk to the instructor BEFORE it is too late.

(7) Make use of the office hours. If needed, there will also be special project discussion hours, to meet as a group and discuss any common problems with the assignments.