COLX 523 Project Proposal

Bingyang Hou, Jae Ihn, Jiexie Kuang, Jinhong Liu, Min Zeng

What is the exact source of the data?

MIT OpenCourseware provides online educational materials for 2506 courses. A course syllabus page containing details of each course is provided by the homepage.

- Information about MIT OpenCourseware - https://ocw.mit.edu/about/
- Link to all courses - https://ocw.mit.edu/courses/
- Link to example course syllabus page - https://ocw.mit.edu/courses/1-00-introduction-to-computers-and-engineering-problem-solving-spring-2012/

What kind of text is it? What are the texts about? Who wrote them?

The course syllabus page contains descriptions about the structure of a university course. It is written by the course instructor and includes some combination of course structure, major topics, course requirements, teaching team, related readings, lecture notes, assignments, exams, etc.

What language is it in? What is the genre/register?

- Language = English
- Genre = Educational resource

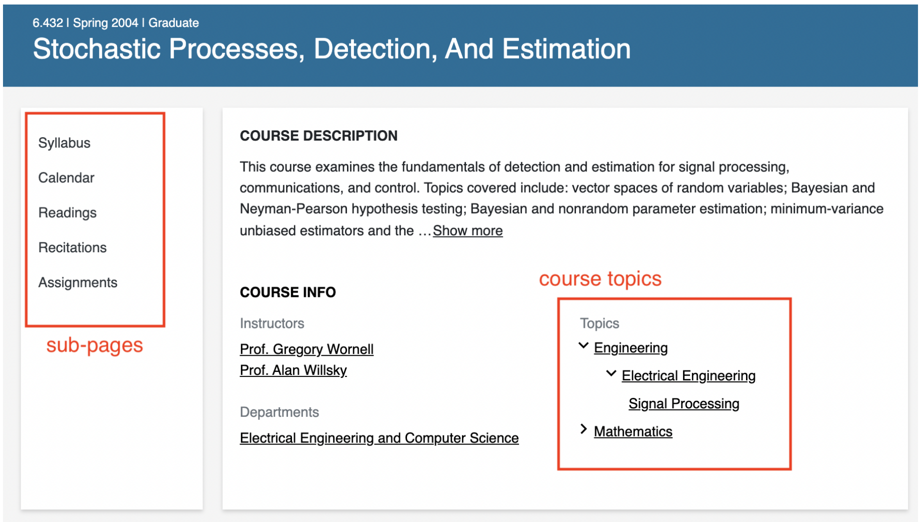
Is there any structure to the corpus you are building (e.g. discussion threads)? Any metadata (e.g. related to author identity)? Will you be targeting a specific kind of text among those available on the site? If so, how will you be filtering the texts to just the kind you want?

The course syllabus page consists of multiple sub-pages that are navigable via a sidebar. Our objective is to annotate “reading resources” across all sub-pages for each course and group them by “course topics”.

The “reading resources” are dispersed in several sub-menus of the course syllabus page (e.g. “Syllabus”, “Readings”, “Related Resources”, etc.). Since not all sub-menus contain a reading resource, we will filter only the relevant sub-menus, and concatenate the contained text to form one long text document for each course.

We will also identify topics assigned to each course. This information is on the main page of the course syllabus. We will keep the hierarchy of the topics provided by the page.

Fig. 1: Example of course syllabus page



6.432 | Spring 2004 | Graduate

Stochastic Processes, Detection, And Estimation

Syllabus

Calendar

Readings

Recitations

Assignments

sub-pages

COURSE DESCRIPTION

This course examines the fundamentals of detection and estimation for signal processing, communications, and control. Topics covered include: vector spaces of random variables; Bayesian and Neyman-Pearson hypothesis testing; Bayesian and nonrandom parameter estimation; minimum-variance unbiased estimators and the … Show more

COURSE INFO

Instructors

Prof. Gregory Wornell
Prof. Alan Willsky

Departments

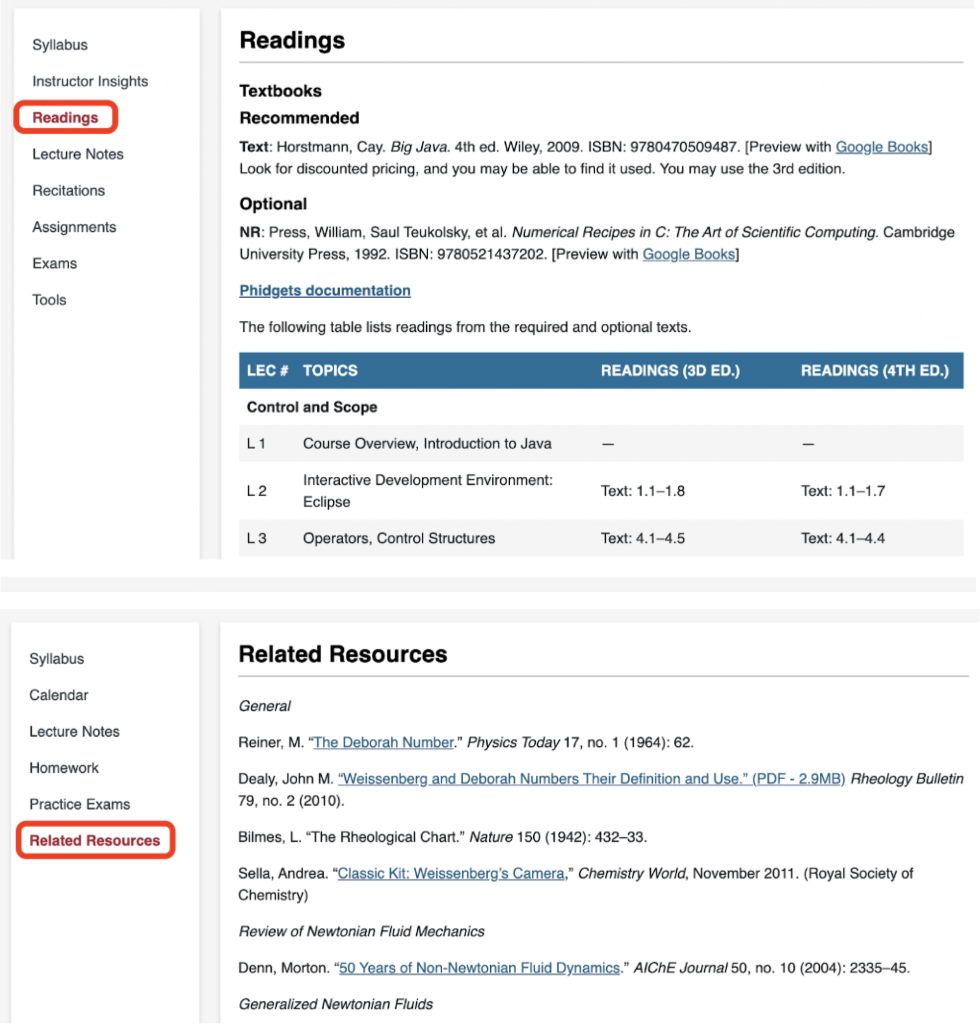
Electrical Engineering and Computer Science

course topics

Topics

- Engineering
  - Electrical Engineering
    - Signal Processing
- Mathematics

Fig. 1: Two examples of relevant sub-pages for reading resources



Syllabus

Instructor Insights

Readings

Lecture Notes

Recitations

Assignments

Exams

Tools

Readings

Textbooks

Recommended

Text: Horstmann, Cay. *Big Java*. 4th ed. Wiley, 2009. ISBN: 9780470509487. [Preview with Google Books] Look for discounted pricing, and you may be able to find it used. You may use the 3rd edition.

Optional

NR: Press, William, Saul Teukolsky, et al. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 1992. ISBN: 9780521437202. [Preview with Google Books]

Phidgets documentation

The following table lists readings from the required and optional texts.

LEC #	TOPICS	READINGS (3D ED.)	READINGS (4TH ED.)
Control and Scope			
L 1	Course Overview, Introduction to Java	—	—
L 2	Interactive Development Environment: Eclipse	Text: 1.1–1.8	Text: 1.1–1.7
L 3	Operators, Control Structures	Text: 4.1–4.5	Text: 4.1–4.4

Syllabus

Calendar

Lecture Notes

Homework

Practice Exams

Related Resources

Related Resources

General

Reiner, M. "The Deborah Number." *Physics Today* 17, no. 1 (1964): 62.

Dealy, John M. "Weissenberg and Deborah Numbers Their Definition and Use." (PDF - 2.9MB) *Rheology Bulletin* 79, no. 2 (2010).

Bilmes, L. "The Rheological Chart." *Nature* 150 (1942): 432–33.

Sella, Andrea. "Classic Kit: Weissenberg's Camera," *Chemistry World*, November 2011. (Royal Society of Chemistry)

Review of Newtonian Fluid Mechanics

Denn, Morton. "50 Years of Non-Newtonian Fluid Dynamics." *AIChE Journal* 50, no. 10 (2004): 2335–45.

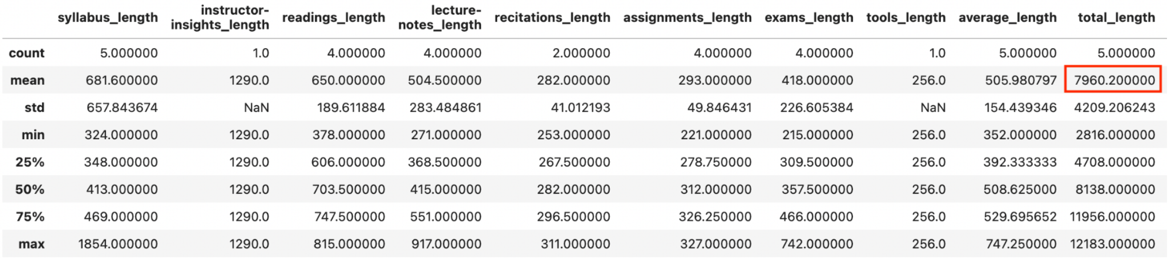
Generalized Newtonian Fluids

How long are the documents, generally? Is there enough data there to create a "Brown-sized" corpus?

MIT OpenCourseware has 2505 registered courses (the webpage states it offers 2506 courses, but upon scraping we have confirmed that there are only 2505–course number 1799 is missing).

The length of each course syllabus page (across all sub-pages) varies depending on the specific course and instructor.

Fig. 3: Length Statistics of Five Courses



	syllabus_length	instructor-insights_length	readings_length	lecture-notes_length	recitations_length	assignments_length	exams_length	tools_length	average_length	total_length
count	5.000000	1.0	4.000000	4.000000	2.000000	4.000000	4.000000	1.0	5.000000	5.000000
mean	681.600000	1290.0	650.000000	504.500000	282.000000	293.000000	418.000000	256.0	505.980797	7960.200000
std	657.843674	NaN	189.611884	283.484861	41.012193	49.846431	226.605384	NaN	154.439346	4209.206243
min	324.000000	1290.0	378.000000	271.000000	253.000000	221.000000	215.000000	256.0	352.000000	2816.000000
25%	348.000000	1290.0	606.000000	368.500000	267.500000	278.750000	309.500000	256.0	392.333333	4708.000000
50%	413.000000	1290.0	703.500000	415.000000	282.000000	312.000000	357.500000	256.0	508.625000	8138.000000
75%	469.000000	1290.0	747.500000	551.000000	296.500000	326.250000	466.000000	256.0	529.695652	11956.000000
max	1854.000000	1290.0	815.000000	917.000000	311.000000	327.000000	742.000000	256.0	747.250000	12183.000000

From **Fig.3**, we can see that:

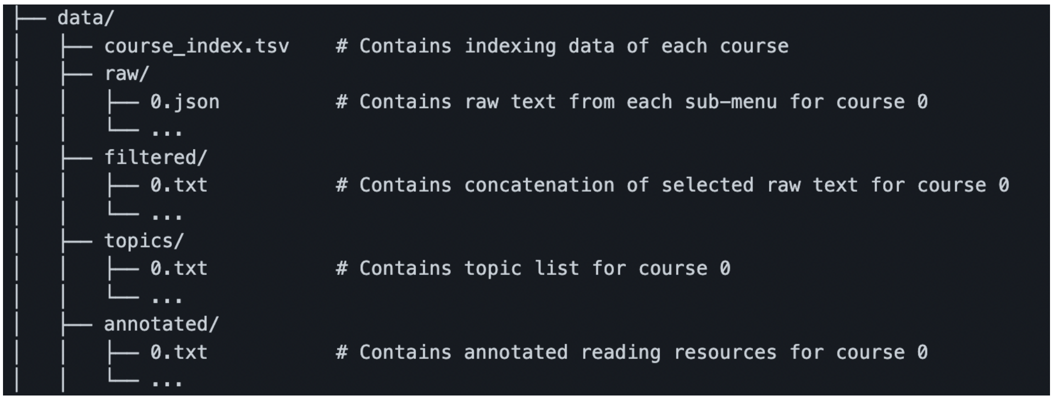
- Minimum length = 2816
- Maximum length = 12183
- Mean length = 7960

Therefore, we have a sizable raw corpus of approximately 2505 x 7960 ≈ 19.9 M tokens.
However, please note that we will be filtering out many sub-pages, so the actual size for this project may be smaller.

What do you have in mind for your annotation of this corpus? In what format are you going to store the corpus and any associated metadata? (JSON? txt? Database?)

1) We will scrape all 2505 courses and store the raw text from each menu for the course as a .json file.
2) We will create a mapping for unique courses to indices.
3) We will discuss which sub-pages we want to filter. After filtering, we will concatenate the raw text into one .txt file for each course.
4) We will extract topics for each course into a .txt file.
5) We will go through the .txt files in (3) to annotate reading resources. These will become .txt files, where each line contains a unique reading resource mentioned in the corresponding course syllabus page.

We expect our file structure to be:



```
├── data/
│   ├── course_index.tsv    # Contains indexing data of each course
│   ├── raw/
│   │   ├── 0.json          # Contains raw text from each sub-menu for course 0
│   │   └── ...
│   ├── filtered/
│   │   ├── 0.txt           # Contains concatenation of selected raw text for course 0
│   │   └── ...
│   ├── topics/
│   │   ├── 0.txt           # Contains topic list for course 0
│   │   └── ...
│   ├── annotated/
│   │   ├── 0.txt           # Contains annotated reading resources for course 0
│   │   └── ...
```

This is subject to change as we develop a deeper understanding of the data.

What makes this corpus potentially of interest? What could it be used for? (Think broadly a lot of corpora have secondary uses beyond the primary annotation.)

For both students and self-taught learners, it is always an important task to be able to find quality reading resources. Our project tasks enable information about reading resources to be collected in one place, so that people can access a comprehensive list of relevant reading resources without having to manually search through several syllabuses.

The way we structure our corpus collection allows for several potential product ideas:

- Recommend readings by course topic
  - e.g. 1) Top 10 books in computer science
  - e.g. 2) Top required readings, for students who want to cover basics
  - e.g. 3) Top optional readings, for students who want to explore more
  - e.g. 4) Bibliography
- Visualize statistical data on readings by course topic
  - e.g. 1) Distribution of types of readings–textbook, paper, chapter, etc.
  - e.g. 2) Gather books by free availability
- Recommend readings connected to other categories
  - e.g. 1) Connect course topics to different careers, then recommend reading resources by career
  - e.g. skill sets that help boost career competence
- Offer comprehensive collection of MIT OpenCourseware resources