

1. What are the names and NetIDs of all your team members? Who is the captain? The captain will have more administrative duties than team members.

Team members: Yuqun Wu(yuqunwu2), Haoyue Bai(haoyueb2)

Captain: Haoyue Bai(haoyueb2)

2. What is your free topic? Please give a detailed description. What is the task? Why is it important or interesting? What is your planned approach? What tools, systems or datasets are involved?

Our topic is designing a crawler for book website. The crawler can scrape similar books and authors based on the initially given website into a local database system, and apply a recommendation algorithm to recommend books for users given queries.

We believe this process contains multiple important concepts and tasks of text information, and developing such a system can help us to have a better understanding about the course.

We plan to use Scrapy package to design a crawler, MongoDB as the local database, and Flask to connect the frontend and backend functions. We might still need to have an investigation about which recommendation algorithm is suitable for our case.

3. What is the expected outcome? How are you going to evaluate your work?

The expected outcome is a frontend as a user interface and a backend system to scrap a book website and make recommendations based on user's preference. The evaluation part is, firstly, the crawler can successfully scrape the books and store them into database. And user can interact with the system through frontend. Besides, we will imitate a user's preference by clicking some books and test if the recommendation system can recommend the books which meet user's preference.

4. Which programming language do you plan to use?

Python and Javascript

5. Please justify that the workload of your topic is at least  $20 \times N$  hours,  $N$  being the total number of students in your team. You may list the main tasks to be completed, and the estimated time cost for each task.

- Implement crawler for book website - 20 Hours
- Implement frontend user interface page - 20 Hours
- Combine a recommendation algorithm into the system - 20 Hours
- Test each part separately - 5 Hours