CourseRatings: A Better Course Evaluation Catalog

Ryan Drapeau
University of Washington
drapeau@*

Emily Gu
University of Washington
emilygu@*

Vimala Jampala
University of Washington
vjampala@*

ABSTRACT
Students at the University of Washington have very few resources to learn about what the classes they could sign up for are like. The University of Washington does provide a course evaluation catalog that summarizes student evaluations of courses. However, the catalog is difficult to use, with users having to rely on their browser's find functionality to search for courses. Additionally, the catalog only lists data from the previous two quarters, making it difficult to observe trends over time. In order to make this data easier to view and understand, we built CourseRatings, an online tool that provides displays evaluations of courses at the University of Washington. Built using React, D3, jQuery, Python, and TaffyDB, CourseRatings allows users to search for, sort through, visualize and compare courses they're interested in. CourseRatings has two goals - to give students and instructors access to the data they need to make informed choices about the classes they take or teach, as well as to present this data in an intuitive and user-friendly way.

Author Keywords
D3, React, JavaScript, Python, database, course evaluations

INTRODUCTION
One of the most stressful times during the school year for a student is registration for the next quarter. Picking the best classes and more importantly, the best instructor for the class could change a student's experience of a course. At the University of Washington (UW), students fill out an evaluation form for the instructors and courses as a whole at the end of each quarter. Currently, UW releases this information in the Course Evaluation Catalog[1] (CEC) to aid students in their search for classes [3].

Motivation
Although the goal behind the catalog is helpful to students, the design of the page and the visualization of the data is not. To find a specific course evaluation, students must first scroll through thousands of previous courses simply listed in alphabetical order. With no search functionality, students resort to using the browser's find functionality which yields poor results much of the time. After clicking on one of the courses, the user is presented with the evaluation data in a simple table. Additionally, since the initial page consists of only a list of evaluations, comparing between different courses becomes a difficult task.

Not only is the design unintuitive, but the data presented conveys little information across to those using the site. When information on a particular class is displayed, all that is shown is the percentage of students that rated the professor "Excellent", "Very Good", "Good", "Fair", "Poor", and "Very Poor" on a particular dimension. It is not immediately obvious whether these ratings mean a class is rated highly or not. The little information that can be gained by looking at this data takes much longer to understand than it should. The site does not directly elicit any average or aggregate information about courses or professors, making it difficult to gain a "big picture" view of the class or any instructor. Additionally, with only the past two to three quarters of evaluation data shown on the UW Course Evaluation Catalog, any data before that time is lost to students.

As a result, we aimed to create a tool that would be useful to students and instructors alike to look at evaluations. Such a tool would give students the ability to create a schedule with classes more suited for them, while also giving instructors the ability to easily understand the feedback they received. Our application reveals deeper insights and user friendly visualizations that would be useful when deciding what courses to take or which instructor to take a course from.

RELATED WORK
There are several existing tools that are similar to CourseRatings at several noteworthy universities, though none of them sets out to achieve the same goals CourseRatings does.

The first is the University of Washington Course Evaluation Catalog, which has been outlined in the previous section in detail.

The second is BerkeleyTime[2], an online course catalog for the University of California, Berkeley [4]. The catalog has

[*] @cs.washington.edu
[1] www.washington.edu/cec/
[2] www.berkeleytime.com/

Submitted to CSE 512 (Spring) as the final paper.

three main pages. The first is a page to search for different courses offered where a course can be clicked to show information about that specific course. The second page is a comparison of grade distributions where the user can choose different courses to compare the grades received. For visuals, a bar chart is used while detailed information, for example average grade is listed. The last page is an enrollment breakdown page. Different courses can also be added to the page and a line chart is used to show the data.

The Berkeley page strives to show a different set of data from what we do. With data on grade distributions and more specific data than we do on enrollment, BerkeleyTime achieves the goal of making course selection simpler in a different manner than UW does. UW on the other hand, uses data from evaluations from students.

The third website is RateMyProfessors[3], which is a website that allows students to rate their professors and their university [2]. This website is similar to CourseRatings, in that it uses student ratings to help other students decide whether or not to take a class. Additionally, unlike CourseRatings, RateMyProfessors also displays comments from students, which provides additional context for the ratings. However, unlike CourseRatings, RateMyProfessors has few ratings per instructor and doesn't support a course search feature. The site is largely focused around providing a means to look up what students are saying about a particular professor. These ratings are also prone to selection bias, since the students who leave reviews on RateMyProfessors are students that have strong opinions about a particular instructor.

The final tool is MyEdu[4], a website targeted towards students and employers [1]. MyEdu allows students to plan their schedules by selecting the courses they are registered for and specifying recurring events (such as work or group meetings). It also lets students browse courses and instructors. Like BerkeleyTime, clicking on a course page in MyEdu shows the average grade distribution for a class, as well as the grade distributions when the class was taught by specific instructors. Similarly, clicking on an instructor page shows the courses taught by that instructor, as well as the average grade distribution of each class when taught by the specific instructor.

Additionally, MyEdu allows students and employers to create and browse profiles. Additionally, employers can create job postings that students can apply to. CourseRatings does not provide this functionality, as applying to jobs and signing up for classes are two separate tasks that have little in common with each other. Additionally, UW has its own student jobs site that is both popular and easy to use.

METHODS

When designing CourseRatings, we used the Course Evaluation Catalog extensively to form opinions of classes and to compare classes. The CEC does a poor job of summarizing the data and displaying it in a friendly manner to the user.

CEC Data and Scraper

Each class is rated along seven dimensions, and the percentage of people who rated the class as "Excellent", "Very Good", etc was displayed. Trying to make sense of these numbers was very confusing, and required going over the table multiple times. Accordingly, we decided to simplify the data CourseRatings presents its users. We reduced the dimensions to the 5 most important ones - Overall, Content, Amount Learned, Teaching and Grading. These were chosen as they are the most populous ratings throughout the dataset. The CEC uses many different surveys to evaluate courses, sections, labs, etc and in total, contains 26 dimensions. The 5 we chose were common throughout a majority of the data. We also realized that most users didn't care about the exact breakdown of votes on each ratings - they cared about the general consensus on the class. Accordingly, we decided to only display the median rating of the class on each dimension, which is shown as the last column in the table on the CEC site. We also wanted to display the percentage of students enrolled in the class who filled out the evaluation surveys, so users can estimate how reliable the data is.

A simple scraper was written using Python in order to capture all of the data found on the CEC site. The CEC separates the catalog into alphabetical sections (A-Z). Each of these sections is processed in parallel to reduce runtime and improve efficiency. Each entry's HTML file is downloaded to a local cache folder if it has not already been downloaded and indexed before. After the scraping has finished, another Python script is ran that parses through the HTML using Python's Beautiful Soup[5] library. Each entry is converted into a vector, where an index into the vector represents that entry's median score along that dimension. These data are outputted to a CSV file, which is then compressed down to 1.5MB (down from >12MB) to speed up payload delivery on the site.

React

The main backbone of the project, other than data, is Facebook's React[6] JavaScript Library. React's abstraction of the DOM allows for better performance when rendering the many tables and visualizations found in CourseRatings. It also allowed for the application to be extremely modular, since all parts are developed into separate components, which made splitting up the work among the team much easier.

Search

One of the first things that strikes most users of UW's Course Evaluation Catalog is that it is very difficult to use it to search for specific classes or instructors, given the lack of search functionality. We wanted our search functionality to both allow users to find specific classes, as well as browse through departments, classes and instructors to view interesting trends. To that end, we considered two different designs for our search feature.

Uniform Search

[3] www.ratemyprofessors.com
[4] www.myedu.com

[5] www.crummy.com/software/BeautifulSoup/
[6] www.facebook.github.io/react/

The first design was to have a single search box at the top of the page. Users could use this to search for departments, course numbers, or instructors. It would return all results that matched the search criteria. For example, a search for the term "ray" would return classes with the word in the title as well as instructors whose names contained it. The main advantage of this design is that it allowed users to perform quick, unconstrained searches. For example, users who wished to see writing classes at UW could just search for "writing" without having to specify a department. However, it had two major drawbacks. The first was that there was no way for a user to specify he or she only wanted search results from a particular department, or search results for a particular type. For example, as in the previous example, if a search query matched both instructors and courses, both would be returned, even if the user was only interested in courses. Similarly, a search for "CHEM" would display not just results from the Chemistry (CHEM) department, but also from the Biochemistry (BCHEM) and Chemical Engineering (CHEME) departments. The second drawback was that typing in the names of classes and instructors took too long. This was exacerbated by users typing in the full name of instructors and courses in order to ensure they didn't receive results they weren't interested in.

Predictive Drop Down Boxes

Our second design was to have three drop down boxes. The first would allow users to specify the department, the second the course code, and the third the instructor. Each drop down list would provide suggestions as the user typed. Additionally, the suggestions for each drop down box would be filtered by what the user had typed in other drop down boxes – each box acted as a constraint on the remaining boxes. For example, if a user specified a particular department, the drop down boxes would only suggest courses and instructors from that department. One drawback of this method is that it prevents partial matches. Users cannot use this search functionality to search for courses across departments that are about "chemistry". On the other hand, this design allows for faster searching, by allowing users to auto-complete names. It also allows users to specify exactly what kind of results they are looking for. This would allow us to tailor the results page to the kind of search being performed. Therefore, we went with this design for the search feature.

Search Results Table

Depending on what the user searched for, information would be returned on either courses or instructors. Each course or instructor would be rated along five dimensions. Therefore, we wanted a visualization technique that would allow users to quickly examine potentially large numbers of search results, each of which would be rated along a number of dimensions. We experimented with several visualization techniques, but all of them fell short when a search result returned a large number of queries. Therefore, we settled on a table. It scaled well to large numbers of results and allowed for easy comparison. We decided to make each column sortable, to allow for easier comparison and exploration. We also decided to color each cell based on the quality of the rating, so that a user would be able to view overall trends in the result. For example, if most of the cells in the search results of a particular department are red, it is a sign that the department does not offer highly rated classes.

RESULTS

The visualizations your system produces and data to help evaluate your approach. For example you may include running times, or the time users typically spend generating a visualization using your system.

DISCUSSION

What has the audience learned from your work? What new insights or practices has your system enabled? A full blown user study is not expected, but informal observations of use that help evaluate your system are encouraged.

FUTURE WORK

A description of how your system could be extended or refined. We have read papers from a number of conferences throughout the course, but if you are having trouble figuring out how to write your paper, take a look at representative papers from the conferences listed above.

ACKNOWLEDGMENTS

We thank Daniel Gorrie, Aaron Nech, and anyone who provided feedback for discussions. We also thank Jeff Heer, Jeff Snyder, and Dominik Moritz for feedback and insights.

REFERENCES

1. J. C. Michael Crosno, Chris Chilek. Myedu, 2008.
2. J. Swapceinski. Ratemyprofessors, 1999.
3. UW. Course evaluation catalog, 2015.
4. A. I. Yuxin Zhu, Noah Gilmore. berkeleytime, 2012.