

Seminar Roulette

Ollie Gardner - 2310049g

December 18, 2020

1 Proposal

1.1 Motivation

Academics around the University of Glasgow often do not have time to trawl through lists of seminars in order to find one that they are interested in and able to attend. Seminar Roulette allows the university community to find seminars which they may not have thought about attending before. By looking at a person's interests and their personal schedule, a recommender system will allow someone to discover academic seminars without any hassle.

1.2 Aims

This project will develop a web application which will allow academics to explore seminars which are taking place at the University of Glasgow. They will have the option to filter out and sort seminars so that they can find a seminar which is tailored to them. Furthermore, they will have the ability to rate past seminars. Using a person's ratings and interests, a recommender system will be utilised to build a profile of them and recommend seminars which might be of interest to them. The usability of the web application will be validated using a System Usability Survey (SUS). Moreover, the reliability of the Seminar Roulette recommender system will be evaluated by conducting experiments.

2 Progress

- Language, frontend framework and database chosen; project will be implemented using Python, Django and PostgreSQL for the backend and ReactJS for the frontend. Material UI will be used on the frontend to build React components based off Material Design principles.
- Conducted an initial user survey with some academic staff in order to create a set of requirements and features which end users would like to see in an application of this nature.
- Wrote up functional and non-functional requirements and prioritised them using MOSCOW.
- Created an initial database schema, basic wireframes and user stories.
- Conducted background research on existing seminar sources e.g. Samoa and EventBrite.
- Initial prototype developed which pulls in seminar data from Samoa and EventBrite. The prototype allows users to find seminars at a time which suits them and can also find a seminar at complete random.

- Developed a recommender system which uses matrix factorization via singular value decomposition to make seminar recommendations to users.
- Carried out a think aloud evaluation on the initial prototype with some of my supervisor's postgraduate students.

3 Problems and risks

3.1 Problems

The following issues have been encountered in the project thus far.

- Seminar data can sometimes be sparse when pulled in from the various data sources. Often, a seminar's title and description shows as TBA or TBC. This can be frustrating for the end user as they have no idea what the seminar will be about.
- Seminars on EventBrite are often uploaded to many different user profiles. It has been tricky to ensure that my script, which pulls in data from EventBrite, encompasses all the seminars which are taking place around the university.
- Deploying the web application to my VM has been problematic at times due to the VM sitting behind the university's firewall. This has made it difficult to install and configure some packages.

3.2 Risks

- Difficult to ensure to that the system will be available and useful to the wider university community. **Mitigation:** using the university's single sign on will enable the web application to be accessed by the whole university community. I will ensure at the start of next semester that I am pulling in seminar data from sources which will interest as many academics as possible.
- Unclear how to connect to a user's Microsoft Exchange calendar in order to show seminars which fit around their schedule. **Mitigation:** will conduct background research into how to use Microsoft's authentication to access a user's calendar events - it is possible that this feature might not be able to be implemented.
- Unclear how reliable the recommender system will be since the rating matrix could potentially be sparse. **No clear mitigation available at this stage.**

4 Plan

4.1 Semester 2

- Week 1-2: act upon feedback and fix issues that arose from the think aloud evaluations conducted at the end of semester 1. **Deliverable:** enhanced prototype which is improved upon from using feedback gathered in the think aloud evaluations.
- Week 3: robustly test the web application and fix any bugs which arise from the testing phase. Ensure the final implementation is polished. **Deliverable:** build upon the current CI pipeline to ensure there is 100% code coverage and code is passing all unit tests.

- Week 4: research into different evaluations that I could carry out to test the reliability of my recommender system and the usability of my application. **Deliverable:** detailed evaluation plan, with participant numbers and list of questions to ask them.
- Week 5-6: conduct final evaluations with participants and also the evaluation of the recommender system. **Deliverable:** quantitative measures of usability and qualitative data gathered from evaluations with participants.
- Week 7-11: write up dissertation. **Deliverable:** multiple drafts submitted to supervisor in the weeks leading up to the deadline.
- Week 11: create video presentation for project and submit it. **Deliverable:** pre-recorded video presentation lasting no longer than 10 minutes.

5 Ethics

This project will involve tests with human users. I have verified that the ethics checklist will apply to any evaluation I need to do. I have already signed and completed the ethics checklist. The evaluations will be carried out anonymously and won't capture any personally identifiable information.