

D4 Personal Reflective Report (F21DG)

Frankland, Matthew
H00241997/mf48

December 22, 2020



A program has been developed throughout the last semester, as part of the F21DG Design & Code group project, to analyse the benefits and limitations of evaluating topic models through quality metrics and perceptual-difference metrics. The purpose of this document is to provide a critical analysis of the work carried out. I will discuss what aspects of our system I am most proud of and those which, on reflection, could have been better; the skills I have developed and those which require further development; as well as critiquing the group methodology for communication, planning and progress monitoring.

The final system was split into four sections: Topic Model Generation, Model Serialisation, Computation of Metrics and the Visual Report. Our group split into three subgroups to work on these sections asynchronously. This helped reduce the time spent on development which was severely limited due to a number of the group, including myself, being on Industrial Placement. Nevertheless, we continued to have weekly meetings, as a whole group, for the purpose of discussing current progress, demonstrating completed work, and prioritising tasks. It was emphasised at these meetings that at any time during the week, if more help was needed in one subgroup, anyone could contact the other groups to ask for help. Each group created separate branches on GitLab which were kept up to date with regular commits, with the master branch containing only final copies of code.

Before anyone started coding, we took time to plan the system and created a series of class diagrams, exemplar data schema and wire frames. This allowed the group to visualise the structure of various system objects, the sequence that these objects would interact in, and what data would be passed between each section. Our system was object-orientated and so these diagrams also helped explain architectural decisions and the system's dynamic behaviour in previous reports. I have not previously had to write many schema or wire frames, as my main specialist area is coding, however this did provide an opportunity for me to develop my User Interface skills.

I was a member of the subgroup tasked with developing the Topic Model Generation and Model Serialisation, and this is the task I will focus my reflection on. The requirements for this section were initially difficult to implement, however, through team work, task management and prioritising tasks I quickly developed a better understanding of our task. We used a shared development environment to work on code as a group which allowed us regular opportunities to learn the Mallet library code together, peer review each others code, catch bugs before pushing to a higher version-control branch, propose improvements and discuss changes. One member of the subgroup would commit to GitLab after each development session to maintain a cohesive workflow.

One area of learning that I am particularly proud of, from this project, is the development of my group based problem solving skills. As a group, we were able to split complex goals into smaller, more manageable, ones that could be completed in parallel. For example, while one person worked on a wrapper for a Topic Model, another could be working on accepting an input configuration file, and another could be working on serialisation of Topic Models. I also learned to be more precise in how I designed my code so that it was highly-adaptable and

future-proof. This meant our code was expandable if more Topic Model data needed to be derived.

Task delegation is also an area I have improved on through this project. Delegating tasks to the right individuals was highly important as it allowed each to focus on producing the best code possible. This was another reason for taking time to plan, before coding began, as the whole group could get a better understanding of each others strengths and weaknesses at a high level, for example coding, user interface, user experience, architectural etc., and at a low level, for example data driven coding, object-oriented coding etc. Based on the different abilities each person had, tasks were assigned accordingly. This helped increase productivity while giving each person an important responsibility, encouraging a greater sense of job satisfaction.

Time management was another crucial skill that I developed further during this project. I was on an industrial placement throughout this project, thus balancing time constraints, with an appropriate amount of delegated work was crucial. Each week I planned the tasks I would complete each day, prioritising the most urgent and important. If tasks mounted up too high, other group members were often willing to fill in for me and I would fill in for them where needed. In this regard we worked well as a team.

If I had the chance to redo the project, I would change how we conducted the integration of our sub-components. I used GitLab histories to get an understanding of changes being made to sections during the week. This led to a few unintentional misunderstandings, particularly in the way our sections would link to the system as a whole. It would quite often be the next weekly group meeting before this became clear and I would have to backtrack to correct the mistakes. It would have been better to have at least one person who was in charge of the overall user experience. This person could have been in charge of the high level overview of the system, planning a strategy for linking the code from each section together, and having a coherent overview of each subgroup thus avoiding the need for backtracking.

Another area of improvement would be for me to give more constructive, actionable feedback focused on an individual's specific work. While I gave feedback, it was quite often general points and not focused on a single piece of work. It would have been better to clearly identify issues and explain directly to the designer how I thought an area could be improved. By doing so others would have been more willing to provide feedback on the work I had completed, resulting in an overall benefit to the group. Any feedback that a member of the group gave influenced our approach to system testing. We broke testing down into three distinct groups so that everyone could share some responsibility for the work.

Our approach to testing was to differentiate 'Pre-Integration', 'Integration', and 'Post-Integration' so as to share responsibility for testing. As our group was working under an agile development methodology, all developers were responsible for writing relevant unit tests for their section, this was the 'Pre-Integration'

testing. Our groups approach to this phase testing worked well as it gave everyone responsibility for the quality of their own work. It also provided the group with a larger library of unit tests than there would have been if unit testing had been left until after integration.

Once each section was complete, we were able to provide a variety of serialised outputs for quality assurance testing. This phase of testing, 'Integration' testing, brought each subgroup back together to check that the integrated module/components work properly using these outputs from their given sections. The final phase of testing, 'Post-Integration', involved individual students conducting a series of tasks to test the instillation and use of the system. Tasks were delegated based on trying to include a wide range of operating systems, and all volunteered to do the work. I conducted the MacOS 'Post-Integration' testing.

After a long period of development and testing, I am very proud of the intuitive system that our group has created. The system is easy to install, has a simple user interface, and is highly efficient. I am also proud of the system's architectural optimisations which were completed, for example, to avoid memory leaks. A future improvement to the system would be to include more troubleshooting tools, especially around error handling. It would also improve the system if adding additional Metrics or data to Topic Models was simpler to do, possibly by adding another user interface.

Reflecting upon my experience working as part of the F21DG Group Project, I have evaluated the areas of my work, and the group's work, that have been successful and highlighted some areas that I can improve upon in the future. My problem solving, time management, and task delegation skills have all improved thanks to my work developing the Topic Model Generator and Model Serialisation and in 'Pre-Integration', 'Integration', and 'Post-Integration' testing. I need to improve on bringing components of a broken down system together through better use of version control, such as GitLab, and giving constructive feedback as work is developing for continuous improvement of a project. I will continue to improve upon the wide range of skills I have acquired during the F21DG Group Project as I seek to continue improving my Software Engineering skills in other industrial projects.