EN 601.447/647: Final Project

Fall 2023
Ben Langmead, TA: Jessica Bonnie, CAs: Molly Li & Sakshi Patil

1 Project description

You will conduct a final research project worth 30% of your class grade. You will work in groups of 4 members. If you are registered for 601.447, your other group members must also be from 601.447. Likewise for 601.647. Your group will submit a one-page project proposal by 10/17/2023, 11:59PM. Your group will give two presentations with each group member participating fully. One presentation will be about halfway through the project (week of Monday, 11/13/2023), and one will be during the finals period. Your group will submit final deliverables by 12/8/2023. Deliverables should be submitted to the the "Final Project Deliverables" section on Gradescope.

1.1 How to approach the project

Pick a topic that interests you. Read papers about what others have done in that area. Since there are topics we won't have covered before the proposal is due, we released a separate document (see "Final project docs" post on Campuswire) with specific ideas for what you might investigate. When reading, consider: what methods have been developed? What are they used for? Are they practical? What are their properties? Make a bibliography and include it in your write-up.

Make a goal. For example: "We will adapt method A to be useful for application B." "We will extend method A to additionally handle cases B and C." "We will implement an idea we have for making method A run faster or use less memory." Pick a goal that plays to your skills. Novelty is nice, but not crucial here. It's fine if your idea comes from a textbook or past paper. It's also fine if other teams are working on a similar idea. It's also fine if your idea doesn't perform better than competing ideas. That's how research goes sometimes. Your grade won't suffer simply because you didn't advance the state of the art. Whatever the ultimate result, your project is a success if you do appropriate experiments to understand why your method performs how it does, and you describe it clearly and comprehensively in your write-up and presentations.

You should not focus primarily on data analysis, though you are encouraged to use real data when evaluating your method. For instance, your goal shouldn't be to use an existing computational method to analyze a particular dataset. Focus instead on **methods**.

The best methods to implement are ones that you can readily evaluate and explain. You might think of a method that seems likely to work well – say, a large language model or some other complicated machine learning model – but which is hard to evaluate (because it's big) and hard to explain (because it's complicated and opaque). We strongly suggest you avoid such methods; they may work well, but it will be hard to find interesting and scientifically relevant things to say about it in the write-up.

1.2 Alternate project emphases

While we expect most projects to pursue a research topic, we are also open to projects with a different focus. For instance, we will accept projects where the primary focus is on the **teaching** of a collection of research topics. For such a project, you should develop a range of supporting materials: tutorials, worked examples, pedagogical visualizations or animations, assessments, etc. It would be ideal (but not strictly necessary) if you could administer your teaching materials to actual students, assessing the students' achievement of learning objectives. If more than one group chooses this direction, we could perhaps assign members from one group be the "students" for the other group and vice versa.

We will also accept projects where the primary focus is on **smart visualization of data structures and/or algorithms**. If so, we will expect you to develop a collection of useful animations, including animations that accept relevant input formats, allow for relevant setting of parameters, and explore relevant images. The visualizations should help to explain and synthesize some complex topic(s) in data structures or algorithms, related to but not necessarily included in what we covered in class. It would be ideal if you could have actual students try your visualizations and give feedback. If more than one group chooses this direction, we could perhaps assign members from one group be the testers for the other group and vice versa.

If you are choosing any of these alternate project emphases, please make this clear in your proposal document. If you choose to switch to one of these alternate emphases later on, after having submitted your proposal, please consult with the teaching team by emailing cs447@cs.jhu.edu.

1.3 Project proposal

Submit a 1-page project proposal to the "Final Project Proposal" section on Gradescope by 10/17/2023, 11:59PM. This gives you time to form teams, discuss topics, and meet with the instructor, TA and/or CA to refine your topic as needed. The format of the project proposal should be:

- List of team members
- Research goals
  - What is the method you want to develop?
    - If you choose one of the alternate project emphases then replace "method" with "visualization" or "teaching unit" as appropriate.
  - How will you evaluate your method?
  - What input data (sequencing reads, genomes, etc) do you need for your experiments?
  - List of at least 3 *milestones*: tasks to accomplish on the way to the research goal
  - List of at least 2 *stretch goals*: things to do if you finish milestones sooner than expected
- References to at least 3 papers related to your chosen topic

The teaching team will give you feedback and may ask you to refine your proposal so that you can get started on the right foot.

1.4 Presentations

Your group will present your project twice, once in the week of Monday, November 14, and once at the end of the semester during the reading period. The final presentation must occur before the end of the scheduled final exam time for this class. The exact final exam time is still not known; we will update this document when we know.

601.447 For both presentations, your group will have a 20-minute Zoom discussion with the instructor and TA or CA about your project. You will start by giving a 10-minute presentation describing your work. Each group member should present for about an equal part of the time. The presentation should use visual aids to reinforce your arguments. Prepare an outline of what you want to say and rehearse it beforehand to ensure you don't go far over the allotted presentation time. A common mistake is to try to cover too much material in a short time. After the presentation, the instructor and TA/CA will ask questions about your work and your individual contributions.

601.647 For both the mid- and final project presentations, your group may submit a recording of the presentation including at least slides and audio. Including video of the speaker is optional, but please indicate on the slides/screen who is speaking at all times. We suggest recording it using something like QuickTime, Camtasia or Zoom and storing/sharing it using OneDrive. It is fine if your presentation is a collection of videos, e.g. if different team members recorded their portions separately. If so, be sure to indicate what order we should watch them in, e.g. in the file names. If you find another good option for recording and sharing, let us know.

Note: If you prefer, you may also request to give your presentation live or over Zoom. Email cs447@cs.jhu.edu to arrange this.

Both 601.447 & 601.647 The instructor and TA will have read your final project document; you don't have to cover everything from the document in your presentation. Use your presentation to cover the main points and the points that are most amenable to visual/oral presentation. For example, if you implemented a complicated algorithm, showing an example of how it works might be a better use of your presentation time than a detailed walk through the algorithm code.

- Group grading criteria
  - Was the project clearly motivated?
  - Were the goals explained clearly?
  - Was the proposed work put in proper context relative to prior work?
  - Was speaking time distributed evenly amongst group members?
  - Was the discussion organized in a way that promoted understanding of the underlying topic?
  - Was the discussion organized in a way that promoted understanding of the proposed approach?
- Individual grading criteria

  - Did the speaker use a clear, audible voice?
  - Was time used appropriately (not too short or too long)?
  - Were visual aids used appropriately?

While we have high expectations of the presentation, we are not trying to mimic a real-world professional setting with proper attire, etc.

1.5 Deliverables

Your group will submit all of your final deliverables by 11:59pm 12/8/2023. Submit to the "Final Project Deliverables" section on Gradescope. Include:

- Source code for all software produced and a detailed README that helps us navigate your code. *We should be able to easily reproduce the results you describe in your report. Tell us how to do this in your README.*
- A write-up of about $N + 5$ pages, where N is the number of people in your group, following the format description in section 1.8 below. Please be concise; do not submit a very long write-up (far more than $N + 5$ pages).
- A detailed (at least 1 paragraph) description of who in the group did what.

1.6 Late policy

The late policy for the final project is separate from the late policy for homework. You can't use homework late days for the final project. You may continue to submit materials up to 48 hours after the deadline, but you will be penalized. We will determine the penalty based on how late you are and how much of the submission is late. Materials submitted more than 48 hours past the deadline will not be considered.

1.7 Source code

Use GitHub (https://github.com) or a similar site for collaborating and for storing your code. But you must submit your actual code (not just a link to GitHub) directly to us. We will grade only code that has been submitted directly.

If you used code from other people, *be very careful to credit them* and provide a link to exactly where you obtained the code in your README. Likewise, if you quote from a paper or website in your write-up, credit the paper or website you are quoting.

We will try to run your software on the undergraduate or graduate Computer Science clusters. These are the same clusters you use for your homework assignments. If your project has exceptional computational needs (e.g. lots of RAM, a GPU or other special-purpose hardware) please let us know ahead of time so we can plan accordingly.

1.8 Write-up

Your write-up should have the following sections:

1. Abstract (1 paragraph): Briefly describe essence of your project
2. Introduction: Why work on this? Why your approach?
3. Prior work: What did you read? What did others accomplish before you?
4. Methods and software: What did you implement? Why? How?
5. Results: How well did your method work compared to others?
6. Conclusions: What did you learn? What should we come away with?
7. Literature cited: What papers are related to your project?

Unlike a typical research paper, your write-up can and should describe everything you did in the project, not just the successful parts. If you spent time thinking about something, researching something or implementing something, even if it ended up not working well and being discarded, go ahead and describe this in the write-up. If there was a method you considered, or started to implement but abandoned, mention this in "Methods and software" and describe why you chose not to go with it.

In the Results section, present a thoughtful evaluation of your method. Compare to other methods if possible. If comparison is only really possible using synthetic ("simulated") data, that's fine. Ideally, your comparisons should involve both synthetic and real data. The following page has suggestions for where to obtain relevant data: `http://bit.ly/CG_GettingData`. If you need more help finding an appropriate dataset, ask us.

1.9 Grading

You will be graded in the following categories. We give rough estimates for point values, but final point values will be decided later.

- Is the submitted code clean, easily navigable and well documented? (15%)
- Can we easily reproduce results in the writeup? (15%)
- Write-up:
  - Abstract and introduction (8%)
  - Related work and literature cited (8%)
  - Methods and software, results, conclusions (29%)
- Presentations (grading discussed above) (25%)