CS 4701: Practicum in Artificial Intelligence
Spring 2019

CS 4701 involves the creation of one semester-long programming project. The goal is to exercise knowledge you acquire from one of your AI courses (CS 4700 or others in our curriculum) through the development of a large-scale system. The major effort of the project should involve AI, as opposed to user-interface design, client-server software, etc. These can certainly be part of your system, but it should not be where most of the work takes place – beware, it is very easy to get sucked into programming effort that is not central to the goal of 4701.

Students should work in teams of 3. Any variance from this number requires prior permission and is rarely granted.

You are strongly encouraged to take 4701 only after you have completed 4700 or one of our other AI courses, such as natural language or machine learning. Those who have no prior experience with AI at the start of 4701 have a very difficult time learning enough quickly enough to have as strong a project as students with such prior knowledge. Please consult with the professor or the TAs if you have any questions concerning this.

Your project will have the following milestones/assignments. 10% of your grade comes from completing the milestones. These milestones are not graded other than for whether they were submitted, although points may be deducted if you do not follow the instructions concerning submissions. PLEASE READ ALL SUBMISSION INSTRUCTIONS CAREFULLY. The other 90% of your grade is the result of your project, discussed further below.

1. Friday, February 15, 1:25pm: **Teammate Selection**: Submit your team information as follows:

   This involves TWO tasks. Each should be done only ONCE by any member of your team.

   a. ONE MEMBER of your team should go to https://goo.gl/forms/tK6WjHBqxfX9n4Zx1 and provide the requested information. It will ask you for a name for your team, the names and NetIDs of your teammates, and a brief description of the general area of your project. The brief description does not lock you in to that for your project. Rather, it is there to encourage you to think about the project as you are assembling your team.
   b. ONE MEMBER of your team should go to Gradescope and do the following:
      - Upload a file with this same teammate information. This can either be a screen capture of the Google form or as a separately prepared file.
      - Link in each of the other team members via Gradescope.

2. Friday, February 22, 1:25pm: **Project proposals**: Submit your proposed project.

   This involves THREE tasks. Each should be done only ONCE by any member of your team.

a. ONE MEMBER of your team should upload a copy of the proposal to Gradescope and link in your teammates.
b. Submit a HARDCOPY of your proposal to my office (Gates 352) by 1:25pm on February 22. I will also accept submission of the hardcopy at the start of the 4700 lecture that day at 1:25pm.
c. ONE MEMBER of your team should go to https://goo.gl/forms/cyRAA6XTuBN1TGNJ3 and provide the requested information.

Your proposal is intended to be short and to the point. It should have the following components:

- Title of your project
- Team name
- Team member names with Cornell NetIDs
- Keywords describing your project. This should include the area of AI that your project involves. It can also include other helpful information, such as, if you're building a game playing program, the name of the game.
- A clear and concise description of what you plan to do:
    - Section 1: What you want to do. This is the meat of your proposal. It is intended to only be a few paragraphs.
    - Section 2: Clearly identify what aspect of AI you are exploring in your project. For example, it might be machine learning, natural language, etc.

Make sure to read this next section carefully.
Evaluation is a key element of your project.

-
    - Section 3: Discuss how you will evaluate your system. THIS IS IMPORTANT! Make sure you have thought about this carefully. How will you know if you were successful in what you attempted? Saying, for example, "I'm going to build a system to play <game>" doesn't address how you will know how well your system plays the game or what elements of your design might have contributed to its success. For example, if you're building a game playing program is there an existing program you will go head-to-head against? (And if so, be realistic about it – for example, it's easier to go head-to-head with an open-source game playing program that you can download and run locally than to do so with a game that you can only play via, say, an iPhone.) Or will you have it play against humans? (And if so, how will you know how good the human players are, to calibrate your results?) If you plan to explore different solution approaches, how will you determine their relative merits? The quality of your evaluation will be an important part of your grade. Note that this is *not* saying that you know what the outcome will be before you even start your project. You might very well create a mediocre system, and that's ok if it shows real effort and you documented how you determined that it was mediocre. Also, be aware that in general testing a system through use by its designers is poor experimental methodology. If you want human subjects you should recruit people other than yourselves, such as others 4701 students using Piazza.
    - Section 4: A timeline for your implementation *and* evaluation. Don't just say "We'll do X first, then Y second, then Z third." Think about how you will break up your project

into pieces, how long each piece should take, which things can be done in parallel versus must be done sequentially, who will do which parts, etc. This timeline is not a contract, it's to help you think about how much you're proposing to do and whether your team has the necessary talents for the proposed project. You should explicitly include time for your evaluation and report writing in your timeline.

- Section 5: Any existing resources (software, data, etc.) you are thinking of relying on. You do not need to reinvent the wheel. If there are software or data resources that you want to use in your project that's ok. It is the work you do beyond these that is graded. Thus, for example, if you propose a project that involves machine learning this is where you would demonstrate that you've thought about where you will find data. Finding existing data is fine, just as is creating your own data resource if need be, depending on the nature of your project.
- Section 6 (If relevant) How this project relates to other work team members may be involved in. Your project can be connected to something else you are currently doing. The only requirements are that (1) the work for the project must be disjoint from what you are doing elsewhere, (2) you disclose that this is the case, and (3) all parties (both this project's teammates as well as those involved in whatever else it is connecting to) are aware of its use in this fashion. It is extremely important that you are clear what is distinct in this project compared to the work you would be doing otherwise.
- Section 7 (If relevant): List any technical papers, websites, etc., you used in formulating your project

This is not intended to be long. There is no explicit page limit for this, but as a rough guideline 1 page will probably be on the short side, 4 pages on the long side. The most important thing, however, is to take the space that you need to provide all the requested information. If you are planning to do a game playing program, for example, and the game is well-known you might not need to spend much time describing it, whereas a project that targets a less well-known game might require more space to explain the game somewhat. That's fine. Similarly, depending on your project you might find that you want to include various figures to explain things, and that, too, might make your proposal longer. That is fine too. If your project is related to something else you're doing, or if you need to take time describing existing software or data resources, you'll also likely have a longer proposal. All this is fine.

You will receive feedback on your proposal. It is not your grade. We want you to succeed with your project and the feedback is to help you do so. You may be asked to submit a revised proposal that addresses certain questions that we have. In some cases this may be because we have concerns about your project, but in other cases it's simply that we needed some additional information to be sure we understand your project and can give you helpful feedback.

On (extremely) rare occasions circumstances have arisen that made a team want to change projects midstream. You cannot do so without permission. Should that wind up the case you would need to prepare a replacement proposal following the guidelines above.

3. Friday, March 29, 1:25pm: **Status Reports:** There will be one status report due about half-way to the due date of the final report.

This involves ONE task. It should be done only ONCE by one member of your team.

   a. ONE MEMBER of your team should upload a copy of your status report to Gradescope and link in your teammates.

   It is natural for your project to morph over time. The goal of the status report is to (1) give details about project elements that weren't apparent at the time of the project proposal, (2) give you a chance to explain how what you're hoping to do has changed, and (2) make sure work is progressing. This status report should start with the project proposal that you originally submitted and directly add to it. Please make it easy to see what was in the original proposal versus what is new text. One good way to show what is new versus what was already there is to use a different colored font for your new text to distinguish it from the earlier proposal text.

   The best case scenario is that things are going exactly as you planned, which would make your status report trivial to prepare. But things *never* go exactly as planned. It is totally ok if things have changed or there's more that you didn't think of at the time of the proposal. This update will most typically include one or more of the following:

   - We didn't realize that doing the part X was so hard, so it's taking us longer to do it and we had to scale back some other part Y.
   - We said we would do X but it doesn't seem to be working so we decided to instead do some modified version of X.
   - We discovered this great software/data resource which simplifies X, which means we're now going to try to do something bigger, X+Y.
   - We learned that the great software/data resource X that we expected to use wasn't available, so we're instead using Y.
   - We thought more about how to evaluate the project, and instead of doing X we now realize that it will be better to do Y.

   Please also provide an updated timeline. It should say which items in the original timeline have been done, and what (if anything) you are changing in the timeline given what you've learned about the project thus far. Similarly, if the planned division of labor has changed, say so. Please also add any additional references to technical papers or websites if relevant.

4. Tuesday, May 14, 4:30pm: **Final Project Reports**:
   You will need submit three things: a team report, an individual report, and your code.

   Group Report: This is the main report for your project.

This involves TWO tasks. Each should be done only ONCE by any member of your team.

a. ONE MEMBER of your team should upload a copy of the proposal to Gradescope and link in your teammates.
b. Submit a HARDCOPY of your proposal to my office (Gates 352) by 4:30pm on March 14.

Your final submission should be a roughly 10-page paper. As with the proposal, this is a guideline that you might very well vary from. For example, your report might be longer if you require many figures. To help you calibrate this guideline, I would guess that 5 pages would like be too short, 15 pages too long.

Your report should include the following:

- Title of your project
- Team name
- Team member names with Cornell NetIDs
- Keywords describing your project
- The body of your report will have two main sections of equal importance:
    - Section 1: Say what you had planned to do, then explain what your project wound up being. Explain what were the key aspects of AI that were the core of your project.
    - Section 2: Explain what assessment you conducted concerning your project. How well did you do? Be clear about both (1) what *questions* you asked, (3) *how* you went about answering them, and (2) what specific answers you got. This will likely be quantitative. Your question(s) might be about how well your system worked (perhaps along multiple dimensions), or it might be about how different elements of the system design contributed to overall success. Your project is not complete if you have not thought about how you can evaluate your accomplishments; the nature of this evaluation is part of the design of your project. Be clear why you think you were successful, or not, or in what circumstances. Sloppy or hand-wavy discussions of performance will significantly negatively impact your grade.
- Make sure to include citations to any technical papers, books, websites, etc., that played a role in the formulation and execution of your project.
- List any other classmates not on your team who contributed to your project. This may be as test users of your system or that you got feedback on your report from them. I STRONGLY encourage you to get feedback on your report from others before you submit it. There is no excuse for a report that has many needless typos or poor writing, and it will significantly affect your grade.

Make sure you read this!

Note what this DOESN'T include: code listings, software architectures, etc. If you're discussing the different modules of your system your report is at the wrong level. Your report should be about the ideas. At this point in your studies I'm presuming that you can architect and code software, and what heft there is in your hacking should be apparent through the ideas and evaluation you conducted.

Individual Report: Each team member should also use Gradescope to submit a short (1-2 page) personal report on the project. You should not submit hardcopy.

This involves ONE task that must be done separately for EACH student.

This report should have three brief sections:

- Section 1: Outline your individual contributions to the project.
- Section 2: Describe what personal lessons were learned from doing the project.
- Section 3: (If relevant) If you played a role in the project of another team, such as serving as a test user or reading their report, you should document this here.

Code: You will also need to submit a copy of your software.

This involves ONE task. It should be done only ONCE by one member of your team.

a. ONE MEMBER of your team should upload a copy of the code to Gradescope and link in your teammates.

You will NOT be graded on the quality of your code. It doesn't even matter what programming language(s) you use. However, we do want to have your code as a record of your work. This will allow us to check, for example, that your team implemented the code and that it does what you reported it does.

Finally, BE AMBITIOUS! It is far better to try for something interesting and get disappointing results than to perform perfectly on something uninteresting. And to be clear, this is about the results being disappointing, not that the effort put into experimental evaluation was disappointing. The amount of effort that your project shows will be an important factor in your grade.

Project Ideas:

Some of you might already have ideas for projects that you would like to tackle – and you are encouraged to pursue them if so – but if you need some inspiration for possible projects, here is a list of sample project topics. *Artificial Intelligence: A Modern Approach* by Russell and Norvig also includes project ideas among the exercises at the end of various chapters that might also provide project inspiration.

- Build a program that plays some game. Important characteristics to consider are:
    - How many players does the game have?
    - Is there a source of randomness, like dice rolls or card deals?
    - Is all the information public, or do players have private information (like their hand of cards)?
    - How easy is it to figure out how good a situation in the game is?

- Machine learning. There are many possible projects that involve machine learning, some of which are listed here. Note that this should be something more than taking an off-the-shelf method and using it on a dataset that you found on the internet. What makes this a project that requires a three-person effort to accomplish? Sample ideas:
    - Make Predictions by Mining Social Media
    - Image Recognition
    - Forecast Financial Markets
    - Learning to Play a Game
- Genetic Algorithms and Evolutionary Programming. This is an area of AI that gets very little attention in typical AI courses that you might very well want to explore in greater depth. Genetic algorithms are a search method that applies when states have structure where “mutations” and “crossover” on their syntactic forms seem to have semantic significance in the domain. Genetic programming has achieved tantalizing results when the objects being searched for have more hierarchical structure, such as learning grammars from sample sentences or learning computer programs by sample I/O pairs.
- A theorem-proving system for some (small) subset of mathematics
- An interesting problem that would involve search. For example, could you write a program to automatically generate crossword puzzles? What information would it require?
- Use reinforcement learning to build a system that learns how to perform some task, such as play a video game like Tetris.
- A “multi-agent system” where each agent has some goals/reward function that it targets in the presence of other agents who (depending on the task) might similarly be working collaboratively on those goals or adversarially to achieve their own goals.