CS3793 n CS5233 – Fall 2020 – Team Project Guidelines

Team Project

This document contains the different guidelines that you will follow for the team project for the AI class for the Fall semester of 2020.

What will you do?

You will be in charge of building a solution to a well-defined problem. The problem you choose is flexible. However, you need to work on a solution, including the concepts discussed in this course.

What are the different sections of the project?

Your final project will consist of the following different sections:

- Problem description -

  This phase is the most important of the project, given that in this stage, you should be able to state what your system will do. Identifying the real-world problem that this system will give solution also needs to be described. The input and output of the system need to be listed.

  The scope of the project should be well defined as well, where it is not too narrow or broad—as an example, building a system to solve the Deep Fake Detection Challenge (DFDC) that is sponsored by Facebook and Microsoft is too broad. Although, an implementation of the available tools/libraries already coded and provided, such as the Chatbot (Watson API) will be too narrow.

  The measures to evaluate the success of your system also need to be defined in this phase. For this, you need to obtain a reasonable size dataset of example input-output pairs, either from existing sources or collecting one from scratch. The size of the dataset will depend on your specific problem.

- Framework -

  To do something interesting, you have to set up the framework. For machine learning tasks, this involves collecting data (either by scraping, using crowdsourcing, or hand labeling). For game-based tasks, this consists of building the game engine/simulator. While infrastructure is necessary, try not to spend too much time on it. You can sometimes take existing datasets or modify existing simulators to save time, but if you want to solve a problem you care about, this is not always an option. Note that if you download existing datasets that are already preprocessed (e.g., Kaggle), then you will be expected to do more with the project. It would be best if you also listed in this section all the HW that will be used for the project.

- Approach -

  Identify the challenges of building the system and the phenomena in the data you're trying to capture. How should you model the task (e.g., using search, machine learning, logic, etc.)? There will be many ways to do this, but you should pick one or two and explain how the methods address the challenges and any pros and cons. What algorithms are appropriate for handling the models you came up with, and what are the tradeoffs between accuracy and efficiency? Are there any implementation choices specific to your problem?

- Literature Review -

Make sure you list, review, and compare different already implemented approaches with the one you propose. That said, no straight-forward efforts already carried out by someone else will be allowed.

- Error analysis -

Design at least two different experiments with a different hypothesis to show the properties (both pros and cons) of your system. Analyze the data and show either graphs or tables to illustrate your point. Which of your hypothesis is approved? Was that the expected result at the beginning of the experiments?

What are the deliverables, and when are they needed to be turned in?

In the remaining of the semester, there will be the following deliverables where you will be able to get feedback about your project.

- Problem Selection (15% points) (2 pages Max) – Due: 10/14

Define the problem statement. Collect some preliminary data, and give concrete examples of inputs and outputs. Implement a lower and an upper bound of the scope of the project and discuss the gap. What are the challenges? Which topics of the class are you implementing(e.g., search, MDPs, etc.) might be able to address those challenges (at a high-level, since we haven't covered any techniques in detail at this point)? Search the Internet for similar projects and mention the related work. You should basically have all the framework (e.g., building a simulator, cleaning data, HW) completed doing something interesting by that point.

- Progress Report (15% points) (4 pages Max) – Due: 11/04

Propose a model and an algorithm for solving the problem. Describe in detail the model and the algorithm. Demonstrate the success or failure of your approach with a concrete example. Add a detailed description of how the different methods apply to solve the problem (what are the variables, factors, states, etc.). You should describe all the tools HW and SW used in your approach. You should also have finished implementing a preliminary version of your algorithm (maybe it is not fully optimized yet, and it does not have all the features you want). Report your initial experimental results.

- Poster (20% points) – Due: 11/18

Your poster should be based on a finished implementation, run a good number of experiments, and done some basic error analysis. In the poster, you should describe the motivation, problem definition, challenges, approaches, results, and analysis. The goal of the poster is to convey the important high-level ideas and give intuition rather than be a super-detailed specification of everything you did (but you should still be precise).

- Final Report (40% points) (5-10 pages) – Due: 11/25

All parts should have completed all the phases described above (problem description, framework, approach, literature review, error analysis).

- Presentation (5% points) – Due: Presentation day

Slides that will be used to show the complete project to all your classmates.

- Group evaluation (5% points) – Due: Presentation day

  A peer evaluation will be filled up by all the members of the team. It will be uploaded individually. An average of the peers' evaluation will be considered to penalize members' involvement in the project.

All deliverables should be submitted through Blackboard by 11:59 pm of the due dates. A single ZipFile should be submitted containing all the files (pdf, peer evaluation, ppt, etc.). For the final submission, you should include a pdf file with the links to the code (compressed in "code.zip") and the data files (compressed in "data.zip"). You should also include a README file with the repo/zip documenting what everything is and what commands are needed for the solution to run.

A file size limit is 20MB per file. If the data does not fit in the file size limit, submit a small but meaningful subset of the data.

Who can be part of the team?

Graduate teams

- 1 team of 3 students
- 1 team of 4 students

Undergraduate students

- Teams of 5 students

All groups need to submit a list of the member through Blackboard the groups by 9/30. Also, you need to create the groups in Blackboard by 9/30. It is preferred if you make your teams by affinity. However, the instructor will assign a team to students with no team after 9/30.

How is the project going to be graded?

The different sections will be graded will be with the following criteria.

- Project description:

  Is the task precisely defined, and is the motivation for the task clear? In other words, does the world somehow become a better place if your project were successful? We will reward projects that are extra thoughtful about how to use AI for social impact.

- Approach:

  Were the lower and upper bounds of the scope of your project well defined, and an advanced method described clearly, well justified, and tested?

- Data and experiments:

  Have you explained the data clearly, performed systematic experiments, and reported concrete results?

- Analysis:

  Did you interpret the results and try to explain why things worked (or didn't work) the way they did? Do you show concrete examples?

- Extra credit:

  Does the project present interesting and novel ideas (i.e., would this be publishable at a good conference)?

Of course, the experiments may not always be successful. Getting negative results is normal, and as long as you make a reasonably well-motivated attempt and you explained why the results came out negative, you will get credit.

Datasets

You can use existing datasets.

- Natural Language Processing Datasets links to many NLP datasets for different languages
- Kaggle for machine learning competitions
- Stanford Large Network Dataset Collection
- SAT Competition Satisfiability problems

Libraries

You can use existing tools for parts of your project. However, it does not mean that the instructor will be the one teaching them to you. When using existing tools, the expectation is that you will do more on other dimensions.

- Scikit-learn machine learning library implemented in Python
- Natural Language Toolkit (NLTK) a set of tools for basic NLP in Python
- OpenCV Python libraries for simple computer vision

Some project ideas:

- Predict the price of airline ticket prices given day, time, location, etc.
- Predict the amount of electricity consumed over the course of a day.
- Predict whether the phone should be switched off / silenced based on sensor readings from your smartphone.
- Auto-complete code when you're programming.
- Answer natural language questions for a restricted domain (e.g., movies, sports).
- Search for a mathematical theorem based on an expression which normalizes over variable names.
- Find the optimal way to get from one place on Stanford campus to another place, taking into account uncertain travel times due to traffic.
- Solve Sudoku puzzles or crossword puzzles.
- Build an engine to play Go, chess, 2048, poker, etc.
- Break substitution codes based on knowledge of English.
- Automatically generate the harmonization of a melody.
- Generate poetry on a given topic