

Project Proposal for CS 175, Winter 2021

Project Title: title goes here

List of Team Members:

Name1, StudentID1, uci_email_address

Name2, StudentID2, uci_email_address

Name3, StudentID3, uci_email_address

General Advice on Writing your Proposal

- Please read the instructions below for each section before you start writing
- A typical proposal should be about 2 to 3 pages in length. 2 pages are fine if you can get all your information into 2 pages, but feel free to go to a third page if needed.
- A useful way to start writing your proposal is to sketch a rough outline of your project with your team members and make sure you agree on the high-level details (statement of problem, types of algorithms you plan to use, data sets, evaluation methods, etc): and then you can start writing the details for each of these in the sections below.
- Feel free to use sub-headings, bullets, paragraphs, etc within each section if it will help make your points clearer
- Figures are often very helpful in describing an idea. Tables can also be useful. (And you can re-use or update any figures/tables again in later reports)
- Don't just cut and paste different sections into the final proposal in a single pass. Generate at least 1 or 2 drafts that everyone in the team reads through and makes comments on, and then make further edits (if needed) to improve the flow and consistency of the document. Writing a good technical document is an iterative process, not just "one pass."
- If different people write different sections make sure the ideas and terminology across the sections are consistent for the reader: this will require checking what other team members have written and correcting potential inconsistencies with other sections.
- In your final submission remove all the text in this template except the numbered section headings.

1. Project Summary

What this section should contain:

- A clear description (no more than 1 or 2 sentences) that clearly summarizes the problem that your project will address, e.g., "This project will use XX methods to predict YY using the Z1 and Z2 data sets, with evaluation using classification accuracy and user studies."
- Make sure this description makes it very clear to the reader what your project is about

2. Technical Approach

- Write a paragraph or two with a clear description of the AI/machine learning methods and algorithms you plan to use in your project. You don't need to know all the details at this stage,

or even all the methods you will use, but you should have some ideas about what you plan to use.

- For most projects you will want to include both at least 1 baseline method and 1 advanced method (e.g., baseline = logistic model with embeddings or BOW as feature input, advanced = RNN + embeddings as input).
- If the system you are building can be thought of as a pipeline with multiple components a useful approach is to provide a figure that illustrates the pipeline with blocks for different components, along with brief descriptions of each component (e.g., the names of algorithms or methods you plan to evaluate).
- Make sure it is clear what your pipeline or system is doing, i.e., what each component will do in terms of taking inputs and producing outputs. Be clear about what the overall input and output is.
- For some projects you may find it useful to have separate descriptions of how the model
- Feel free to add a reference or two (see the class Web site for suggestions, or do a search in Google Scholar using appropriate keywords).

3. Data Sets

- Briefly describe what data set(s) you plan to use in the project (if data set(s) are an important part of your project). Include references to the data (e.g., a URL) if you can.
- If for example you are doing document classification, you can describe for example how many documents are in the data set, average document length, how many classification labels. If you are using multiple data sets you could put this type of information in a table.
- You could for example whether you plan to work with data that is already tokenized and already has a predefined vocabulary or whether you plan to investigate different tokenization methods and explore different vocabularies.
- Make sure that at least one person on your team has successfully downloaded the data and taken a look at it, just to check that its available and that generally it matches your expectations.
- You could for example include a figure or two in this section, e.g., a histogram of document lengths or some other summary information about your data. If you are using Python for creating figures with data, the Python seaborn package is very useful for generating plots.
- You can change your data sets during the project if you need to, but you should have identified at least one data set to work with by the time you submit the proposal.

4. Experiments and Evaluation

- Provide a brief and clear description of how you will evaluate the results of your project, e.g., accuracy for classification, precision-recall for document ranking.
- NOTE: Lecture 6 on Monday will discuss different evaluation approaches – so you might want to plan to finish this section after Monday’s lecture
- Be creative in how you think about evaluation: instead of just looking at a single evaluation metric such as accuracy or error rate (for example) you can also evaluate your system by looking (manually) at the types of errors it makes (see Lecture 6 on Monday for some tips on this).

- For some projects (such as language generation, e.g., for chatbot response) you may have to do some user studies for evaluation, e.g., present users with results from Algorithm A and Algorithm B, using the same input data for each algorithm, without telling the user which algorithm is which, and have them select the one they prefer – if you plan to do this it would be good to think through how many users you plan to have participate, at what stage of the project you would do this (e.g., you could do it once to get initial feedback and later for a final evaluation), and so on.
- If your project involves unsupervised learning tasks like clustering or topic modeling you may have to do some research to see how evaluation is done on these tasks.

5. Software

- Provide a list of the major pieces of project software that you expect to use, divided into 2 sets: (1) publicly-available code, and (2) code will write yourself. (Students in the past have found it useful to use a table in this section)
- The list of what public software you will use will probably be incomplete at this point (which is fine) since you may not know yet about all of the software that might be relevant to your project.
- My expectation is that most students will use Python, given that we have been using Python in class and there are many useful publicly-available tools for text analysis in Python. However, if you prefer to use a language such as Java that is ok too - please indicate this clearly in this section.
- You may also want to use a tool such as Github to coordinate code development on the project among your team – if you have not used Github before this would be an excellent opportunity to learn about it.
- Note that each team member is expected to write a non-trivial component of code for the project (this could be a specific machine learning algorithm, some code for cleaning noisy data during preprocessing, a script for creating pipelines and running experiments, code for visualizing/displaying results, etc). Each student on a team will need to be able to point to their specific code in the progress report and in the final report. Given this, it is important that in the early weeks of the project that team members identify “who is doing what” – this can change as the project evolves. In this section you describe what code the team will write (your best estimate – this may get updated later as you get into the details of your project) – and in Section 7 below but you will provide your current best guess about who will do what.

6. Milestones

Provide a brief list of milestones. For example, since the project will span roughly 6 to 7 weeks of the class (weeks 4/5 to 10), you could break your milestones into a list of 3 intermediate phases:

- Weeks 4, 5 and 6
- Weeks 7 and 8
- Weeks 9 and 10

For example, much of the data gathering and preprocessing and coding (development and test) could happen in the earlier weeks, and much of the experimentation and evaluation in the later weeks. Note that you have a progress report due at the end of week 7.

7. Individual Student Responsibilities

Summarize briefly what each student will be primarily responsible for in the project. For example, you might write something like this

Name 1: will write and test the code for Algorithms 1 and 2, will integrate components A and B into the pipeline, will assist in doing experiments and interpreting results, will assist in writing project reports

Name 2: will acquire the data sets to test the algorithms, will preprocess the text data (e.g., define the vocabulary for the algorithms), will implement baseline Algorithm 3 and integrate it into the pipeline.

Name 3: will supervised integration of all the components from Names 1 and 2 into a pipeline, will write the scripts for evaluating the accuracy of the algorithms, will take the lead in writing project reports.

[Note these are just suggestions – you can and should organize responsibilities in whatever way makes sense – and inevitably as the project progresses these responsibilities may need to be changed as some tasks may take much more time (or much less time) than originally expected.]