

# CS224n: Project Milestone Instructions

February 2019

Each team (both default and custom) hands in **one** project milestone, which is worth 5% of your grade. The milestone should help you make progress on your project, practice your technical writing skills, and receive feedback on both. This document specifies what information you should include in your milestone – it applies to both default and custom projects.

## 1 Milestone contents

Ultimately, your final report will be written in the same style as a NLP / Deep Learning research paper. For the milestone, we ask you to write a preliminary version of some sections of your final report. Producing a high-quality milestone is time well-spent, because it will make it easier for you to write your final report. You might find that you can reuse parts of your project proposal in your milestone. This is fine, though make sure to act on any feedback you received on your proposal.

Your milestone should be a PDF created using the NeurIPS 2018 LaTeX template<sup>1</sup> – make sure to turn on the camera-ready setting i.e. remove the row numbers. It should be written in a way that a fellow CS224n student could understand. Your milestone should be **2-4 pages**<sup>2</sup> (not including references). It should contain the following sections (though you can use a different structure if you prefer):

**Title:** The title of your project (you can change this later).

**Team members:** Provide the names and @stanford.edu email addresses of all of your team members.

**Abstract:** An abstract should concisely (less than 300 words) motivate the problem, describe your aims, describe your contribution, and highlight your main finding(s). Given that your project is still a work-in-progress, it's OK if

---

<sup>1</sup><https://www.overleaf.com/latex/templates/template-for-neural-information-processing-systems-nips-2018/zxpdbfpqgxss>

<sup>2</sup>We expect that default project milestones will generally be shorter than custom project milestones. If you're doing a relatively straightforward default project, make sure to thoroughly describe your approach and experimental details, and thoughtfully discuss your results.

‘your contribution’ and ‘your findings’ are things you’re still working on.

**Approach:** This section details your approach(es) to the problem. For example, this is where you describe the architecture of your neural network(s), and any other key methods or algorithms.

- You should be specific when describing your main approaches – you may want to include equations and figures (though it’s fine if you want to defer creating time-consuming figures until you write your final report).
- You should also describe your baseline(s). Depending on space constraints, and how standard your baseline is, you might do this in detail, or simply refer the reader to some other paper for the details. Default project teams can do the latter when describing the provided baseline model.
- If any part of your approach is original, make it clear (so we can give you credit!). For models and techniques that aren’t yours, provide references.
- If you’re using any code that you didn’t write yourself, make it clear and provide a reference or link. When describing something you coded yourself, make it clear (so we can give you credit!).

**Experiments:** This section contains the following.

- **Data:** Describe the dataset(s) you are using (provide references). If it’s not already clear, make sure the associated task is clearly described.
- **Evaluation method:** Describe the evaluation metric(s) you used, plus any other details necessary to understand your evaluation.
- **Experimental details:** How you ran your experiments (e.g. model configurations, learning rate, training time, etc.)
- **Results:** Report the quantitative results that you have found so far. Use a table or plot to compare multiple results and compare against baselines.
  - If you’re a default project team, this means you should **make at least one submission to the dev leaderboard, and report the F1 and EM score(s)** in this section. Make it clear whether you are on the non-PCE or PCE leaderboard.
  - Comment on your quantitative results. Are they what you expected? Better than you expected? Worse than you expected? Why do you think that is? What does this tell you about what you should do next? Including training curves might be useful to discuss whether things are training effectively.
  - You don’t need to report any qualitative results (‘analysis’) in the milestone, though you can if you like.

**Future work:** Describe what you plan to do for the rest of the project, and why. You can include stretch goals if you like.

**Additional information:** Include the following information, as necessary:

- **(Custom projects only) Mentor:** Give the name of your mentor. If you have an external mentor, additionally list the CS224n staff member who graded your project proposal.
- **(Optional) External collaborators:** If you have any collaborators who are not CS224n students, list them.
- **(Optional) Sharing project:** If you are sharing your project between CS224n and another class, indicate it here.

**References:** Your references section should be produced using BibTeX.

## 2 Grading and feedback

Your milestone will be graded on the following criteria:

- **Progress:** Has the team made good progress on the project? You should have done approximately half of the work of your project.
  - For default projects, you should have implemented a significant part of your final model(s). As a minimum, you should have obtained dev leaderboard scores that **improve** on the provided baseline.
  - For custom projects, as a minimum, your milestone should show that you have setup your data, baseline model code, and evaluation metric, and run experiments to obtain some results (assuming you are doing a typical model-building project). Other than this, ‘good progress’ depends on various factors (e.g., whether your model is implemented from scratch or based on an existing codebase).
- **Understanding:** Does the milestone show a strong understanding of its problem, tasks, methods, metrics, and research context?
- **Writing quality:** Does the milestone clearly communicate what you’ve done and why, providing the requested information, to an appropriate level of detail (given the page limit)?

You will receive some brief feedback on your milestone. Feedback may contain:

- Helpful suggestions for your project (e.g., try a particular method, read a particular paper).
- Warnings about your project plan (e.g., if your plans are too ambitious or not ambitious enough).

- How you could improve your technical writing (e.g., adjustments to clarity, level of detail, formatting, use of references).

If you are doing a custom project, your milestone will be graded by your mentor. If you have an external mentor, it will be graded by the same CS224n staff member who graded your proposal (though you should also ask your external mentor for feedback on your milestone).

### 3 Improving your technical writing

Technical writing is an important skill in this class, in research, and beyond. It's well worth spending time developing your ability to communicate technical concepts clearly. Here are some resources which might help you improve your technical writing:

- *Tips for Writing Technical Papers*, Jennifer Widom.  
<https://cs.stanford.edu/people/widom/paper-writing.html>
- *Write the Paper First*, Jason Eisner.  
<https://www.cs.jhu.edu/~jason/advice/write-the-paper-first.html>
- *Writing in the Sciences*, a Coursera course.  
<https://online.stanford.edu/courses/som-y0010-writing-sciences>
- **The Hume Center for Writing and Speaking** offers drop-in and appointment-based sessions to get help from a writing tutor.  
<https://undergrad.stanford.edu/tutoring-support/hume-center/see-tutor/what-expect/faq-students>
- Stanford Engineering's **Technical Communication Program** provides courses and individual consulting.  
Info: <https://engineering.stanford.edu/students-academics/technical-communication-program>  
Writing tips: <https://stanford.edu/class/ee267/WIM/TechWritingTips.pdf>

Here are some other things you can do to improve your technical writing:

- Look carefully at several NLP / Deep Learning papers to understand their typical structure, writing style, and the usual content of the different sections. Model your writing on these examples.
- Think about the NLP / Deep Learning papers you've read (for example, the one you summarized for your proposal). Which parts did you find easy to understand and why? Which parts did you find difficult to understand and why? Can you identify any good writing practices that you could use in your technical writing?
- Ask a friend to read through your writing and tell you if is clear. This can be useful even if the friend does not have the relevant technical knowledge.

- You can get help with your technical writing in Office Hours! TAs are happy to discuss best practices for technical writing. You can even bring a draft of your project milestone (or final report) for the TA to give feedback on. As previously announced, you cannot ask TAs to ‘verify’ the content of your milestone/report, but you can ask TAs to give feedback on the clarity/structure/style of your writing. Depending on the busyness of office hours, you may need to choose just one section of your writing for the TA to look at.

## 4 Submission instructions

Submit your PDF proposal on Gradescope (this time we have two separate Gradescope assignments – one for default and one for custom projects). *Make sure to tag all of your team members* – only tagged team members will receive credit.