CS224N: Project Milestone Instructions

Last updated on February 16, 2025

Each team hands in one project milestone, not exceeding **two pages** in length (excluding figures and references), which is worth 6% of your overall grade. Milestones that include significantly more than two pages of written, non-reference content will be penalized. The milestone should ensure you make progress on your project, practice your technical writing skills, and receive feedback on both. This document specifies what information we expect you to include in your milestone. It applies to both default and custom projects. We provide a milestone **template**:

https://www.overleaf.com/read/hkhkpcycnvjn#ba186b

In addition, we encourage you to take a look at **sample reports** from last year. Note that unlike the previous year, we restrict the maximum page limit to be two pages this year.

https://drive.google.com/open?id=18Lt8cBaCE9g7HpqhjPINL26Y78Ek3i2U
https://drive.google.com/open?id=1sfkHtKyvrdtA5OnKDx_wgvCNHMdeBy5G

1 Milestone contents (max 2 pages)

Your final report will be written in the same style as a research paper (though obviously much shorter than most research papers). For the milestone, we ask you to write a preliminary version of 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 may find that you can reuse parts of your project proposal in your milestone. This is fine, although you should make sure to incorporate any feedback you received on your proposal.

Your milestone must be a PDF file created using the provided template above. It should be written in a way that a fellow CS224N student can understand by assuming knowledge already taught in class. Note that this means technical terminology you've picked up while learning topics specific to your project should be concisely defined, but you don't need to explain material covered in class or the default project handouts (e.g. what Sonnet is). Your milestone can be maximum 2 pages (excluding figures and references); **note that this means each section must be concise!** As we use the milestone to assess progress, we

encourage you to spend less space re-introducing your approach and more space on **what you've accomplished and what new plans you've made since the original proposal**. Your milestone must contain the following sections:

Key information.   Your PDF milestone should have the following information:

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

- **Team member names**: List the names and `@stanford.edu` email addresses of all of your team members.

- **Custom or Default Project**: Indicate which you are doing.

- **Sharing Project**: If you are sharing this project between CS224n and another class, you **must** indicate it in the milestone. Otherwise, you can leave this part out.

- **TA Mentor**: This is the TA that graded your proposal.

- **(Optional) External Mentor**: If you are doing one of the projects proposed by Stanford AI Lab and community members, write their name(s).

- **(Optional) External Collaborators**: If you have any collaborators who are not CS224n students, list them.

Abstract.   Your abstract should motivate the problem, describe your goals, and highlight your main findings. Given that your project is still in progress, it is okay if your findings are what you are still working on. Writing an abstract is very difficult, as it should be concise, high-level, and convincing. Keep it to a few sentences!

Approach.   This section details your approach to the problem. For example, this is where you describe the architecture of your models, methods, or algorithms.

- Please be specific when describing your main approaches. You may want to include key equations and figures (though it is fine if you want to defer creating time-consuming figures until the final report).

- Describe your baselines. Depending on space constraints and how standard your baseline is, you might do this in detail or simply refer to other papers for 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. For models and techniques that are not yours, provide references.

- If you are using any code that you did not write yourself, make it clear and provide a reference or link. When describing something you coded yourself, make it clear.

Experiments.   This section is expected to contain the following.

- **Data**: Describe the dataset(s) you are using along with references. Make sure the task associated with the dataset is clearly described.

- **Evaluation method**: Describe the evaluation metric(s) you used, plus any other details necessary to understand your evaluation.

- **Experimental details**: Please explain how you ran your experiments (e.g. model configurations, learning rate, training time, etc.). This should concisely provide technical details, not a narrative.

- **Results**: Report the quantitative results that you have so far. Use a table or plot to compare multiple results and compare against your baselines.

  - If you are a default project team, this means you should **make at least one submission to the dev leaderboard for paraphrase detection and report the accuracy**.
  - 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 do not need to report any qualitative results in the milestone, though you can if you want.

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

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, your milestone should show that you have setup your data, baseline model, and evaluation metric as well as have run experiments to obtain preliminary results (assuming you are doing a typical model-building project). However, for many projects, 'good progress' depends on various factors (e.g. whether your model is implemented from scratch or based on an existing codebase), and you should discuss with your mentor what a reasonable target is.

- **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 brief feedback on your milestone. Feedback may contain:

- Warnings about your progress (e.g. whether your plans are too ambitious or not concrete enough).

- Suggestions for your project (e.g. regarding particular methods or related work for reference).

- Feedback on your technical writing (e.g. adjustments to clarity, level of detail, formatting, use of references, etc.).

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 (you are highly encouraged to ask your external mentor for feedback on your milestone).

3 Tips for 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*, 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.[1]

- Stanford Engineering's Technical Communication Program provides courses and individual consulting.[2][3]

---

[1] `https://undergrad.stanford.edu/tutoring-support/hume-center/see-tutor/what-expect/faq-students`

[2] `https://engineering.stanford.edu/students-academics/technical-communication-program`

[3] 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 papers to understand their typical structure, writing style, and the usual content of the different sections. Model your writing on these examples.

- Revisit the NLP papers you've read (for example, the one you summarized for your proposal). Which parts did you find easy or 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 it is clear. This can be useful even if the friend does not have the relevant technical knowledge.

- As always, TAs are happy to discuss best practices of technical writing. You can bring a draft of your project milestone to get feedback on. Specifically, we encourage you to ask TAs to give feedback on the clarity, structure, and style of your writing.

4 Submission instructions

Submit your PDF on Gradescope. Please follow the same submission instructions for project proposal here. Make sure to tag all of your team members – only tagged team members will receive credit.