COMS4705: Project Report Instructions

Last updated on September 23, 2025

Each team submits one project report, which is worth 35% of your overall grade. This document specifies what information you should include in your report. For this report, we **require** that you create your PDF file using the LaTeX template provided in this link:

https://www.overleaf.com/read/ptqghknzrksp#433129

We will deduct points for using any other LaTeX template. In addition, we encourage you to take a look at sample reports from the past three years of the Stanford CS224N course, which can be found here:

https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1214/project.html
https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1204/project.html
https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1194/project.html

1 Report contents (6-8 pages[1])

Your final report should be written in the same style as a NLP / Deep Learning research paper, and written in a way that **a fellow COMS4075 student could understand.** Your report should be 6-8 pages (excluding references). Keep in mind that longer is not necessarily better; clear, **concise** writing will be rewarded!

Below we describe the expected content of the final report. Sections marked with (**Required**) must be included. You may use a different structure for the other sections if you prefer.

Key information. (Required)

- **Title**: The title of your project.

---

[1]We expect that the lengths of project reports will vary for each project. If you're doing a relatively straightforward project, make sure to thoroughly describe your approach and experimental details, and thoughtfully discuss your results.

- **Team member names**: List the names and <UNI>@columbia.edu email addresses of all of your team members.

- **(Optional) External Collaborators**: If you have any collaborators who are not COMS4705 students, list them.

- **Mentor**: Write the name of your COMS4705 staff mentor here. If you also have any external mentor at Columbia, add their name(s) as well.

- **(Optional) Sharing Project**: If you are sharing this project between COMS 4705 and another class or as part of research outside of the course, indicate it here.

Abstract. (Required) An abstract should concisely (less than 300 words) motivate the problem, describe why it is difficult and existing approaches fail, describe your contribution, and highlight your main finding(s). It is often useful to write your introduction last after other sections are completed.

Introduction. (Required) The introduction section explains the problem, why it's difficult, interesting, or important, how and why current methods succeed/fail at the problem, and explains the key ideas of your approach and results. Though an introduction covers similar material as an abstract, the introduction gives more space for motivation, detail, references to existing work, and to capture the reader's interest. Near the end of the introduction, you should also succinctly state your contributions or main findings. This guide from Jennifer Widom provides useful tips on how to structure an introduction: `https://cs.stanford.edu/people/widom/paper-writing.html#intro`.

Related work. (Required) This section helps the reader understand the research context of your work, by providing an overview of existing work in the area.

- You might discuss: papers that inspired your approach, papers that you use as baselines, papers proposing alternative approaches to the problem, papers applying your methods to different tasks, etc.

- This section shouldn't go into deep detail in any one paper (for example, there probably shouldn't be any equations) – instead it should summarize groups of papers, explain how the papers relate to each other, and how they relate to your work.

- Attempt to demonstrate, as you review the literature, limitations or motivations that point to why *your* work is a nice next step, or useful replication, or promising analysis (or otherwise, if your work doesn't fall into these categories!).

After the introduction and related work, you will have several sections constituting the body of your report. You have the freedom to decide on the exact structure of the body. Below, we provide some suggestions on what sections to include.

Approach. This section details your approach(es) to the problem. For example, this is where you would describe the architecture of your neural network(s) or your novel algorithms. For projects that aren't describing a novel methodology, this section might describe in more detail whatever technical tools (e.g. probing methods, proof techniques, interpretability algorithms, etc.) you'll use in your experiments.

- You should be specific when describing your main approaches – you probably want to include equations and figures.

- 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.

- 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!).

- As you're setting up equations, notation, and the like, be sure to agree on a fixed technical vocabulary (that you've defined, or is well-defined in the literature) before writing and use it consistently throughout the report! This will make it easier for the TAs to follow, and is nice practice for research writing in general.[2]

- If your approach is particularly complex, we encourage you to create a visual diagram to help explain each component, and you should reference parts of the diagram throughout your written description.

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. Being precise about the exact form of the input and output can be very useful for readers attempting to understand your work, especially if you've defined your own task. In particular, you should provide an example or a few

---

[2]In experimental work, this could mean giving a specific name to each method, each dataset, each baseline; it could also mean making consistent use of mathematical notation where appropriate.

3

examples of inputs and expected/ground truth outputs to make the task clear to the reader. [3]

- **Evaluation method**: Describe the evaluation metric(s) you use, plus any other details necessary to understand your evaluation. Some projects will have clear metrics from prior work on given datasets, but we realize that other projects will define their own metrics. If you're defining your own metrics, be clear as to what you're hoping to measure with each evaluation method (whether quantitative or qualitative, automatic or human-defined!), and how it's defined.

- **Experimental details**: Report how you ran your experiments (e.g. model configurations, learning rate, training time, etc.). If you used configurations or hyperparameters from prior work, you should reference the paper. Alternatively, if you chose values for a particular reason, it can be helpful to explain that motivation.

- **Results**: Report the quantitative results that you have found so far. Use a table or plot to compare results and compare against baselines. [4]

  - 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 that tell you about your approach?

Analysis. Your report should include *qualitative evaluation.* That is, try to understand your system (e.g. how it works, when it succeeds and when it fails) by inspecting key characteristics or outputs of your model.[5]

- Types of qualitative evaluation include: commenting on selected examples, error analysis, measuring the performance metric for certain subsets of the data, ablation studies, comparing the behaviours of two systems beyond just the performance metric, and visualizing attention distributions or other activation heatmaps.

---

[3]If your task involves exceedingly long inputs or outputs, such as summarization or long-form question answering tasks, you can include truncated examples (making clear that you did so) or include examples in an Appendix to reference in the body of the paper.

[4]Some analysis-centric projects will not have model-based baselines to compare against; in this case, consider whether there are simpler analysis methods than your proposed method which might achieve the same goal. It would be great to evaluate them, or at least discuss why they are insufficient compared to your method. Even in analysis projects, it's important to ask whether we could be achieving similar insights using simpler ("baseline") methods. So, if you don't have a baseline analysis method to compare against, discuss some alternative methods of analysis that one could use, and why (mathematically, linguistically, or otherwise) you believe your proposed method is superior.

[5]For some analysis-centric projects, it might seem odd to have a separate analysis section. It's up to you whether this section is included explicitly, but it might be a good idea. Your analysis method hopefully has some quantitative evaluation, and you likely had to come up with it. In this section, consider giving examples of your analysis method in action on a given input, or graphs of the dataset; in general, attempt to give the reader intuition and insight into what your model is doing beyond the "topline numbers" described and discussed in the results section.

- The *Practical Tips for Final Projects Lecture Notes* has a detailed section on qualitative evaluation – you may find it useful to reread it.

Conclusion. (Required)   Summarize the main findings of your project, and what you have learned. Highlight your achievements, and note the primary limitations of your work. If you like, you can describe avenues for future work.

Team contributions   We ask that you include a brief summary of what each team member did for the project (about 1 or 2 sentences per person). We will read these descriptions. For almost all teams, it will have no effect (i.e. team members all receive same grade), but for teams with considerably unequal contribution, we may investigate and/or give different grades to team members.

References. (Required)   Your references section should be produced using BibTeX.

Appendix (optional).   If you wish, you can include an appendix, which should be part of the main PDF, and does not count towards the 6-8 page limit. Appendices can be useful to supply extra details, examples, figures, results, visualizations, etc., that you couldn't fit into the main paper. However, your grader *does not* have to read your appendix, and you should assume that you will be graded based on the content of the main part of your paper only.

2   Improving your technical writing

As a reminder, the *Milestone Instructions* included several resources to 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.[6]

- Stanford Engineering's Technical Communication Program provides courses and individual consulting.[7][8]

---

[6]https://undergrad.stanford.edu/tutoring-support/hume-center/see-tutor/what-expect/faq-students
[7]https://engineering.stanford.edu/students-academics/technical-communication-program
[8]Writing tips: https://stanford.edu/class/ee267/WIM/TechWritingTips.pdf

You can use these, and previous feedback you've received, to improve your technical writing.

Always remember to be precise, use consistent technical terminology, and define terms that are clear to you now but aren't known to the average COMS4705 student.

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.

3   Grading and feedback

Your project report will be graded holistically, taking into account many criteria: originality, performance of your methods, complexity of the techniques you used, thoroughness of your evaluation, amount of work put into the project, analysis quality, writeup quality, demonstrating strong understanding, etc. Your final submission should show that you made an effort to incorporate all feedback you received on your proposal and milestone report from your mentor.

Your report will be graded by two staff members, whose scores will be combined into your final score. You will also receive some brief feedback on your report. Your COMS4705 staff mentor will be one of the graders.

4   Code

We ask you to submit your code as a zip file (up to 1MB) to Gradescope.

- **Do** include all project code written or adapted by you.

- **Don't** include the whole source code for off-the-shelf packages that you used without adapting (e.g. CoreNLP or PyTorch).

- **Don't** include model checkpoints or data.

Your code will not be graded—we collect it so that we can investigate honor code issues if necessary.

5   Submission instructions

To summarize, here are the instructions:

- Submit your report to Gradescope under [Final Project - Report] **Make sure to tag all of your team members—only tagged team members will receive credit—, and only submit one Gradescope entry per team**.

- Submit your code to Gradescope under [Final Project - Code].

The due date is and teams can spend up to three late days on the project report.

6   Posting reports online

All final reports will be posted on the COMS4705 website. If you *do not* want your report to be published online, please let us know in a private Ed post, and we won't upload it.