CS-552: Project Report Instructions

Last updated on June 1st, 2023

Each team submits one project report, which is worth 30% of your overall grade. This document specifies what information you should include in your report. For this report, we recommend that you submit a PDF generated using LaTeX in either the ACL or NeurIPS styles. We will accept alternate formats, but they are not preferred.

Report contents (4-8 pages)

Your final report should be written in the same style as a NLP / Machine Learning research paper and should be 4-8 pages (excluding references).

We've provided a sample section structure for you, though you should feel free to use a different structure if it is a better fit for your narrative.

Title

You should come up with a catchy title for your project. You should also provide the author list of your project team and your team name.

Abstract

Your abstract should concisely (less than 300 words) motivate the problem, describe your aims, describe your contribution, and highlight your main finding(s).

Introduction

The introduction 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.

Related work

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 (e.g., there shouldn't be any equations). Instead it should explain how the papers relate to each other, and how they relate to your work. This is not a section to copy-paste your reviews from Milestone 1. Those papers can

serve as the basis of your related work section, but you should synthesize what was learned into a roughly half a page section. 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!).

Approach

This section details your approach to the problem. For example, this is where you describe the architecture of your system, and any other key methods or algorithms. You should be specific when describing your main approaches – you probably want to include equations and figures. You should describe in your approach both how you implemented the reward model and how you implemented your final system.

When writing equations and other notation, be sure to agree on a fixed technical vocabulary (that you've defined, or is well-defined in the literature) before writing. Then, use it consistently throughout the report.

Experiments

This section contains the following.

- **Data**: Describe the dataset(s) you are using (provide references). 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. If there are legal or ethical considerations to the data used, discuss it here.
- **Evaluation method**: Describe the evaluation metric(s) you use, plus any other details necessary to understand your evaluation. 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.
- **Baselines**: You should also describe your baseline(s)
- **Experimental details**: Report 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 results and compare against baselines. Comment on your quantitative results. Are they what you expected? Why do you think that is? What does that tell you about your approach?

Analysis

Your report can include qualitative evaluation. You should 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.

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 behaviors of two systems beyond just the performance metric, and visualizing attention distributions or other activation heatmaps.

Conclusion

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.

References

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 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.

Technical Writing

Please follow these recommendations for good technical writing:

- Be precise and use consistent technical terminology. Define terms that are clear to you but may not be known to the average CS-552 student
- 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 literature review). 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.

Helpful Links:
https://cs.stanford.edu/people/widom/paper-writing.html
https://www.cs.jhu.edu/~jason/advice/write-the-paper-first.html

Grading

Your project report will be graded holistically, taking into account many criteria: originality, 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. Overall performance of your method will be a **small** component of the overall evaluation. We're more interested in the thought and innovation that went into your project than whether it ended up getting the best score.

Models

We ask you to submit your final generative model. The model should be able to generate proper answers given a question. You should have the freedom to choose any model structure (that is generative of course), learning algorithms, and training data for training your generative model. We also ask you to submit the reward model you used to train your generative model. Be aware that the reward model is still compatible with the evaluation script we provided in Milestone 2.

Your generation model does not have to handle multi-interaction prompts, although if you would like to implement such a model you are welcome to. You will only be evaluated on one turn prompts.

Code

We ask you to submit your code to the GitHub classroom. 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). Your code will not be graded. We collect it so that we can investigate course integrity issues if necessary.

Next, we ask you to submit a JSON file consisting of the answers for the testing questions generated by your generative model. With the answers, we ask you to also submit a running script for accessing your generative model and using it to generate answers for test questions. The answers from this script should match exactly to your submitted answers.

It's very important for this script to be reproducible so we can grade you, therefore please also prepare a `requirements.txt` file specifying the package versions, as well as a `python.txt` file to specify the python version.

Team contributions

We ask you to submit 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 the same grade), but for teams with considerably unequal contribution, we may investigate.

Submission instructions

To summarize, here are the instructions for submitting your final deliverable. You need to commit to your final github repository all of the following files:

- **Answers** to the example prompts in `prompts.json`, that you can save in a json file named `answers_<team-name>.json` This file should be a list of dictionaries with 2 keys "guid" and "model_answer". The "guid" is to specify the question id and the "model_answer" key should be your model's generations. The "answer" key in the `prompts.json` file is just to guide you what the solution of the question is. However,

your model's answer does not have to match this. You are welcome to prompt the model for these questions as you want. Make sure the script below handles this so we get the exact same answer.

- **Script** to get this json file output `gen_script_<team-name>.json`
- **Reproducibility files.** The `requirements.txt` file as well as the python version in `python.txt` that helps us run the above script. Make sure that your script is reproducible by testing from an environment from scratch.
- **Your final generation model.** We are flexible about the format of this model as long as you load it correctly in the script to generate answers.
- **Your final reward model.** We are less flexible about this model's format, please follow the same format from M2. But feel free to have it different from what you submitted in M2.
- **Datasets** you curated for both the final reward model and the final generation model. Please name them `reward_dataset_<team-name>.<any-format>` and `gen_dataset_<team-name>.<any-format>` We are flexible about the format you submit these datasets in.
- **The final report PDF** as we have detailed above. Please call it `final_report_<team-name>.pdf`

The due date is 11:59 PM on Sunday, June 18.

Publication policy

All final reports will be shared with the course instructors whose courses you designed a chatbot for.