CS 4650/7650
Project Final Report Guidelines
Due: **Apr 23 at 11:59 PM (anywhere on Earth, UTC - 12:00)**

Final Report

In the final report, students will present the outcome of their research project. The report should contain a summary of the project goal, a specification of the final models and analysis used in the project, the results of the testing, an explanation of how the results fit into related work, and suggestions for future study if any.

Report Content:

1. Abstract (0.25 pts)
    - 1 paragraph
    - **Briefly summarize the goal, methods, and results of the research project**.
    - Provide a link to the code repository. e.g. "the code is hosted at the following link: `github.com/user123/project456`". See "Code repository" below (6) for details.
    - Update this accordingly and do not directly copy the content from the proposal or midway report.
2. Introduction & Related Work(1.25 pts)
    - 2-3 paragraphs
    - **What is the motivating problem for your research project?** What is the goal of the project? If it has changed since the project began, how has it changed and how close is it to the original goal?
    - **Related Work:** Please cite 3-5 related studies and explain their contribution to the problem space (e.g. "*Yang and Prasad (2018) found that attention-based models outperform HMM models for part-of-speech tagging*"). Compared to prior work, how is your project trying to address these gaps? What's your contribution compared to the previous work?
    - Update this accordingly and do not directly copy the content from the proposal or midway report.
3. Methods (4 pts)
    - 3-4 paragraphs
    - **Data (0.25 pt)**
        i. What data did you use for the project? If you did not do so in the midway report, please provide a **table of summary statistics** for the data to help us understand the scope of your project, e.g. number of documents, average document size, unique number of labels. Please also provide several short examples of the data, e.g. for sentiment analysis, a post labelled for positive sentiment and negative sentiment.

ii. If you have changed your data since the midway report, what data have you collected and how is it different from the previous data?

- **Models/Analysis (3 pt)**
  i. **What NLP/ML model did you propose or extend? How did you adapt to address the problem?** Please describe your model in detail. If necessary, provide a figure to show your model architecture. **(2 pt)**
  ii. Or what analysis/testing did you propose to address the problem? What insights could this kind of testing provide into the problem? If necessary, provide a figure to show your analysis pipeline.

- **Baseline Models (0.75 pt)**
  i. What baseline models did you choose to use to compare your model or tests against? Please pick at least two baselines for the comparison with your model or tests and describe them in detail. You may find it useful to consider models from prior work or an original model if appropriate for the task, e.g. a logistic regression classifier.
  ii. If your major goal is interpretability of the model, demonstrating your interpretation across various similar models will help you ground the results of your main model or interpretability method. You should build and compare several models.

4. Results (8.5 pts)
   - 10-12 paragraphs
   - Experiment setup (0.75 pt)
     i. What's your data/model configuration details? E.g., you can include the details like the size/ratio of train, development and test set; the learning rate; optimizer.
     ii. If part of your results required annotation, how many annotators did you recruit, what was the prompt for annotation (e.g. "label all Named Entities in the following text"), and what was the agreement score among annotators?
   - Result comparison (7)
     i. How well did your models perform on a given task? If you are using a standard dataset, report performance on the standard test split to compare your performance with prior work. Use significance testing (e.g. t-tests) to determine significant differences in performance across models.

ii. How well did your models perform in comparison to at least several baselines?
iii. If relevant, which hyperparameters did you test in your model and why?
iv. If your research problem required ablation (e.g. "TF-IDF+word2vec" versus "TF-IDF only"), what features did you choose to change and why? How did the different ablations compare in terms of performance?
v. If you are performing an extended analysis, how do the different components of the analysis fit together? What hypotheses were you trying to test with your analysis, and did you prove or disprove the hypotheses? If the analysis reveals conflicting results (e.g. model A outperforms B but only on certain sub-tasks), what are some ways to explain or diagnose the differences?
vi. If your major goal is the interpretability of the model, you should show as much evidence of your interpretation as possible, e.g. visualization, analysis of intermediate outputs, and control tasks.
vii. **It's OK to have negative results**. If you found negative results, what do you think might have led to these results? How would a different model or test have achieved positive results instead? You may find it useful to include diagnostics including: a plot of training loss; a plot of test performance across different training schemes (e.g. more/less training data), hyperparameter tuning results.
viii. Please include a brief error analysis. This may include features such as a confusion matrix, a table of common errors made by the model, feature weights, or a picture of attention scores over example text. What kinds of errors does your model make, are there any examples, and why do you think it made them?
ix. If you are doing a generation related project, please include some generated examples and their comparison with the ground truth and baseline models.

- Work Division (0.5 pts)
  - **i. Which team members were responsible for which part of the results?**
- **Writing (0.25 pt)**
  - i. Please double check your writing to fix any grammar errors, and proof-read your report before the submission. You may get a penalty if the writing is poor.

5. Conclusion (0.25 pts)

- 2-3 paragraphs
- What are the low-level and high-level conclusions that we can draw from your work? E.g. low-level would be "contextualized embeddings consistently improve QA performance on factoid questions", high-level would be "contextualized models capture a wider range of semantic patterns in typical questions."
- How do your results compare with the related work in the problem space (e.g. "in contrast to Yang and Prasad (2018), we find that attention-based models do not outperform HMMs in part-of-speech tagging")? If you find different results from related work, what factors do you think could explain the difference?
- If someone wants to continue your work, what are some potential future directions for the project? This may include further model development, model analysis or data collection.

6. Code repository (0.75 pt)
   - To provide evidence of the work that you have completed, you must provide a link to the public Github repository.[1] If your project is too big to store on Github, please create a folder on another service such as Box.
   - Your repository must include a `README` file that explains the **directory structure**, i.e. where to find the code for the model, and the **installation instructions**, e.g. what packages to install (preferably include `requirements.txt` if your project is in Python).
   - We expect your code to work. In the `README` file, please provide **instructions to run the code** to generate the main results of the paper. For example:
     "To generate the POS tagging accuracy results, run the following command from the `main` directory:
     `python run_tagger.py --mode test`"
   - To save space, you may choose to not host the full data and saved models in your repository. If you do this, please note it in the `README` file and explain how to run the code with a subset of the data or models.

Writing Specifications:

1. The report should be **at least 7 pages** and **at most 8 pages** in length, including figures and tables but not including references. Do not exceed 8 pages. You should be able to explain your project's main results in a maximum of 8 pages. Reduce unnecessary details and highlight the main contributions of your project.

[1]Tutorial for Git: https://product.hubspot.com/blog/git-and-github-tutorial-for-beginners

2. The report format should minimize whitespace, e.g. no excessive whitespace before or after figures.
3. To stay organized, you should add subsections to your paper. E.g. in "Results," you might include separate subsections for "Model tuning" and "Feature ablation."
4. The report should not include copy-pasted code. We can investigate code in the provided repository if necessary. If necessary, please explain in writing or in math what your code does.
5. The report should be written in **ACL 2020 style** (in LaTeX or Word), which is available here: http://acl2020.org/downloads/acl2020-templates.zip.
6. Overleaf template available here: https://www.overleaf.com/latex/templates/acl-2020-proceedings-template/zsrkcwjptpcd
7. The report should be submitted as a PDF file, on Canvas.
8. The report should be submitted by **one** team member.
9. The report should be named in the following format:
   team_[ID]_final_report.pdf
   The team with ID 1 should name the file team_1_final_report.pdf.

Writing Advice:

1. Do not wait to begin writing until the last minute. You should be adding to the report throughout the project to keep the details of your data, methods and results up to date.
2. Rather than writing sentence by sentence, it may help to begin the report as a bulleted list of points to cover, which you can fill in later.
3. When writing your report, you may find it useful to look at reports from similar courses, such as the NLP course at Stanford.[2] These reports will give you an idea of the level of detail and the writing style that we expect for your report.

[2]See the "Best" project reports here: http://web.stanford.edu/class/cs224n/project.html