# Final Project for CS 4650 (Georgia Tech)

Instructor: Wei Xu

*Please do **not** share this document outside the class as it contains information on an unpublished research idea from Dr. Sanja Stajner.*

There is no final exam for the class. We hope that the final project will give you the flexibility to choose what you want to work on with the time you have, and enjoy some teamwork.

Students **may choose their own topic and project ideas.** Two potential/example ideas are provided on the next page of this document.

While it is not a requirement, you are encouraged to think of making novel contributions to NLP (e.g., new dataset, new application, new model, interesting data analysis, etc.) or NLP+X research (X can be areas you have prior research experience, e.g., vision, data visualization).

**Lecture slides on course project:** https://cocoxu.github.io/CS4650_spring2022/slides/lec13-seq2seq3-project.pdf

**Team Size:** Students may do final projects in teams of 2-4 persons. Alternatively, you could do the project solo, but please email the instructor (include your project idea) to get permission first.

**Report & Oral Presentation:** You will submit a report (up to 4 pages using ACL template; this page limit includes everything, tables, figures, references, etc) with a link to your code/data by Apr 29 11:59pm. **You will also be giving a 5-min oral presentation (over Bluejeans) of your project at the final exam time (Apr 29 2:40pm)**. Besides the interestingness and solidness of your work, the quality of your writing will also be a factor we will consider in grading. Here are some example project reports: https://web.stanford.edu/class/cs224n/project.html

**Contribution:** Please submit a short paragraph that states the individual contribution of each group member. All members in the group are expected to receive the same grade, except in very rare situations.

**External collaborators:** You can work on a project that has external (non CS4650 student) collaborators (e.g., your undergraduate research advisor), but you must make it clear in your final report which parts of the project were your work. If you choose to work on the provided research idea, please also acknowledge Dr. Sanja Stajner who provided the idea in your report.

**Shared project with other classes:** You can share a project between CS4650 and another class, but we expect the project to be accordingly bigger, and you must declare that you are sharing the project in your report.

**Prizes:** You will learn some new knowledge! We plan to give out 1-2 best course project awards (no cash value, but you can list it on your CV).

**Submission Instruction:**

1) 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.
2) Submit your code (and data, if the size is small) to Gradescope under [Final Project - Code].
3) If you are a multi-person team, submit a brief description of team contributions to Gradescope under [Final Project - Team Contributions]. We will read these descriptions; but we expect that for all teams, this will have no effect (i.e., team members all receive the same grade), unless there is a very rare extreme situation with considerable unequal contributions.

**The due date is 11:59PM EST on Friday, April 29.** Due to grading deadline constraints, there will be no late days allowed. So, please plan accordingly.

**Idea #1. Developing a better Dialog System than the one in Project 3**

Starting pointers:
https://cocoxu.github.io/CS4650_spring2022/slides/lec17-dialogue.pdf

**Idea #2. Analyzing Text Simplification Corpus**
*This research project idea is provided by Dr. Sanja Stajner (stajner.sanja@gmail.com).*

Starting pointers:

1) Manually annotate a portion of (complex-simple) sentence pairs from Wiki-manual corpus (https://github.com/chaojiang06/wiki-auto/tree/master/wiki-manual; described in details in this ACL 2020 paper), for whether or not they contain the below-mentioned transformations, or if there are any quality issues (e.g., errors). Have separate labels for whether or not the complex sentence contains the four obstacles (one label per each obstacle: passive voice; conditional verb; hidden verb; distant subject, verb, and object) and then separate labels for whether or not the simple sentence contains the four obstacles.

   Focus on verb and syntax simplifications as per the following four lines in Table 2.1 in this PhD thesis:
   - Use active voice instead of passive
   - Use the simplest form of a verb (do not use conditionals)
   - Avoid hidden verbs (verbs converted into a noun)
   - Keep subject, verb, and object close together

2) Develop a system for automatically detecting these obstacles and measuring the precision and recall on the manually annotated data (Hint: the first three would require only a Part-of-Speech tagger, the fourth requires a parser; we recommend using the Stanford CoreNLP tool for both).