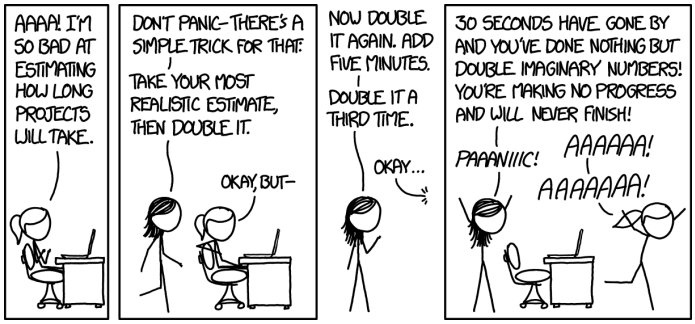
# Final Project

## 1 Overview

The focus for the last few weeks of the semester will be to propose and complete a final project of your own design.




You will collaborate in **groups of 4 or 5** individuals, preferably within the same lab section[1].

You may (and are encouraged to!) form groups of your choosing. Please email Prof. Caplan with your group members **before October 28th at 11:59 AM** Eastern time along with a group name[2]. This is a hard-deadline. If you do not email me by then with your group then I will create/assign a group for you.

**Final Project Dates and Deliverables**

- A **pre-proposal** (∼1 page) is due **November 2** ..... (5%)
- A **full proposal** (∼4 pages), including adjustments based on my comments and a concrete timeline, is due **November 9** ..... (10%)
- Weekly (**Nov 11, Nov 18, Dec 2**) checkpoint demonstrations in lab (to me and/or classmates) ..... (10%)
- Final presentations will take place on **December 10, 9am-12pm** ..... (25%)
- A final paper and working project materials (e.g. all code and data) is due **December 15** (this is a college-deadline: no extensions) ..... (50%)

**If you think something's missing from the description provided here, or there's anything you are not certain of, don't hesitate to ask!**

[1] exceptions can be made of timezone accommodation or other similar extenuating circumstances

[2] this can be anything you want, so have fun, otherwise I'll assign you something cheesy like the 'Wuglet Research Institute'

## 2 Project Ideas

You are encouraged to construct a project of your own design. Both as an exercise in creative engineering, but also because you'll have a lot more fun doing something you are all interested in.

Your project must be novel work related to NLP — it should go beyond what we have covered in the course in terms of assignments or core lecture materials — but otherwise you have a huge amount of flexibility in the style and content of the projects you might consider.

One key piece of advice I have is to make sure the goal is tangible and realistic. For example, many potential projects related to deep learning have failed/will fail not because they require too much coding (with standard libraries they may involve significantly less), but because it is notoriously difficult to debug and manage extremely large data sets (e.g., working with image captioning systems). Look for papers related to your idea to ensure it is tangible. If it is difficult to find related work, treat it as a warning sign. Usually, it means it isn't a suitable NLP task, but it could also mean you simply don't yet know the right terms to search for, or need to flesh out your idea more.

The basic ingredients of a project are as follows:

1. A formal definition of the problem and a motivation for while it is an interesting challenge for natural language processing. A literature review of past approaches to the problem.
2. A commented implementation of the simplest possible solution to the problem. For instance, this could be a majority class baseline, a random baseline, or an implementation of a reasonable baseline published in the literature.
3. A commented implementation of at least one extension that attempts to improve on performance above the baseline. You don't need to success in achieving improved performance, but you will need to do some (at least exploratory) analysis to understand why your system failed if it does.
4. An evaluation script that can be used to automatically score different models. The evaluation code should provide the output reported in your final presentation / paper.

### 2.1 Sample and External project resources

- **Duolingo STAPLE shared task**: *"Machine translation systems typically produce a single output, but in certain cases, it is desirable to have many possible translations of a given input text. This situation is common with Duolingo (the world's largest language-learning platform), where some learning happens via translation-based exercises, and grading is done by comparing learners' responses against a large set of human-curated acceptable translations. We believe the processes of grading and/or manual curation could be vastly improved with richer, multi-output translation plus paraphrase systems.*

*In this shared task, participants start with English prompts and generate high-coverage sets of plausible translations in five other languages. For evaluation, we provide sentences with handcrafted, field-tested sets of possible translations, weighted and ranked according to actual learner response frequency. We will also provide high-quality automatic translations of each input sentence that may (optionally) be used as a reference/anchor point, and also serves as a strong baseline. In this way, we expect the task to be of interest to diverse researchers in machine translation, MT evaluation, multilingual paraphrase, and language education technology fields."*

  - Project idea: translations are ranked according to popularity. What makes one sentence more popular than another? Is it fluency/perplexity? Frequency of individual words? Similarity to the prompt? As far as I know, this isn't something that's been looked at before!

- Modern statistical parsers rely on large, painstakingly hand-annotated corpora in order to train their parameters. However, because patterns of language vary between genres, parser performance suffers when training and testing sets are drawn from different domains. Unfortunately, in practical applications, in-domain "gold standard" labeled data is often unavailable and would be prohibitively expensive and time-consuming to collect. Therefore, developing methods of *parser adaptation* has the potential to enhance the usefulness of standard parsers for application to non-news data. One idea would be to use "lexical substitution": what if you replaced works unknown to the parser (i.e., undertrained) with words having similar semantic/syntactic distributions; since semantic similarity models are trained on unlabeled data, this would be a way to leverage the statistical signal present in vastly more data than a parser can realistically be trained on.
- Here is a list of many, many **past NLP student projects from Stanford**
- Here is a list of a bunch of **past NLP student projects at MIT**
- Check out recent work from the **Association for Computational Linguistics (ACL)**

### 2.2 Shared tasks

Shared tasks are a great project choice. Even if the task has already 'ended' it's a good source of ideas and pre-curated data (often with evaluation scripts already ready to go as well!). Here's an assortment of possible shared tasks to check out:

- Morphology:
  - `https://sigmorphon.github.io/sharedtasks/`
  - `http://morpho.aalto.fi/events/morphochallenge/`
- Machine Translation: `http://www.statmt.org/wmt19/translation-task.html#download`
- Dialogue modeling: `https://www.parl.ai/`

- Question Answering: `https://stanfordnlp.github.io/coqa/`
- SemEval tasks:
  - `http://alt.qcri.org/semeval2019/index.php?id=tasks`
  - `http://alt.qcri.org/semeval2020/index.php?id=tasks`

## 2.3 Collecting your data

Since most of the systems we've built in this course are data-driven, it's important to have your data ready to go towards the beginning of the project. You should collect all of the data that you'll need for your final project and split it into three sub-corpora: **Training data, Development data, and Test data**.

If you are basing your term project on a shared task, then usually the data will be collected already, and usually it will be divided into a standard training/dev/test split. If it's already assembled and split — great! You're ahead of the game. If you're not doing a shared task, then you may need to assemble your own data. A good way of creating your own training/dev/test split is to divide the data into chunks that are sized around 80%/10%/10%, where you want to use most of the data for training. **It's important to ensure that the same items don't appear in more than one of the splits!**

If your data is very large, then your repository can simply include a sample of the data and give a link to a Google Drive that contains the full data set.

If you have interest in working with Twitter data, you can fairly easily scrape some yourself: see this **Twitter API tutorial from Prof. Wei Xu**

### 2.3.1 Example Project Datasets

A broad amount of data exists for NLP projects. If you have a topic in mind and don't know where to search, get in touch with me and I can help point you in the right direction. For now, I'll link below to some sets of potential datasets for a range of projects:

- Common tasks and state-of-the-art methods in NLP: `http://nlpprogress.com/`
- Public domain NLP datasets: `https://github.com/niderhoff/nlp-datasets`
- Sentiment analysis: `https://www.kaggle.com/bittlingmayer/amazonreviews`
- Sentence inference: `https://www.kaggle.com/stanfordu/stanford-natural-language-inference-corpus`
- Fake news: `https://www.kaggle.com/mrisdal/fake-news`
- Topic modeling: `https://github.com/tdhopper/topic-modeling-datasets`

- Discourse acts: `https://github.com/google-research-datasets/coarse-discourse`
- Coreference resolution: `https://github.com/google-research-datasets/gap-coreference`
- Semantic role labeling: `https://www.cs.upc.edu/~srlconll/`
- Semantic parsing from natural language questions: `https://www.cs.utexas.edu/users/ml/geo.html`

Also do read through the various shared tasks in Section 2.2 since you can use their data sets veen for tasks that weren't originally intended by the organizers.

# 3 Proposal

## 3.1 Pre-proposal

Before midnight on **November 2**, you will submit your pre-proposal (a 1-page abstract) **as a PDF** via your Git repo. I will use this to provide advice in lab on November 4th. You pre-proposal should include at least:

- Your group name and the names of all team-members
- A project title
- A description of your project (at least a paragraph)
- A set of question that you still need to answer to flesh out the details of the project. I will attempt to answer these / point you in the right direction to the relevant resources, but this is also something you will be evaluated on, so please put some thought into it ahead of time.

## 3.2 Full Proposal

Before midnight on **November 9**, you will submit a full proposal (depending on the structure this might be about 4 pages plus the timeline) **as a PDF** via your Git repo. Your full proposal should include the following:

- The project title, group name, and list of all team-members
- A central hypothesis. What is the main question you would like to answer (i.e. your goals).
- Provide some background for the problem. You should cite at least 6 papers, and provide a summary for each paper (1 large paragraph is sufficient, somewhere between a third and a half a page per paper). For more details on this "literature review" see the *"how-to"* box below.
  - Background should include both some grounding of the particular problem as well as grounding in the particular NLP solution. For instance if you are building a language

model to rank different candidate translation output, you'll need to ground your approach both in the literature on MT as well as the literature on algorithms for language modeling.

  - Make sure to have a section at the end where you properly provide references for all works cited

- What is the baseline system you will compare against? Do you need to implement it yourself or is there an existing library you can pull from? if the later, which ones?
- What is/are the central algorithm(s) or solutions you are proposing? Again, do you need to implement these from scratch or can you pull components from existing libraries (if so, which ones)?
- What data set(s) are you using? Are you creating them for a real-world problem, or are you using a standard repository data set? Either way, be specific[3]
- What experiments and what type of analysis do you plan to execute? What do you expect the results to be? You will need to outline at least some speculative analysis you can perform to understand your result in the case that your solution fails.
- A week-by-week timeline. What are the concrete progress milestones that you plan to hit? Be specific and realistic — don't think you need to claim that you will produce a ground-breaking state-of-the-art results, but do write a couple sentences to outline your planned progress for each week. There will be check-ins on November 11, November 18, and December 2 where you will update me on which goals you have completed.
  - The timeline also needs to identify which team-member is the **point person** who assumes responsibility for their component of the project. Depending on the specifics of each project, all components may end up being shared components, but you still need to lay out who takes responsibility for managing what sub-parts.

[3] data problems can make or break a project!

**Literature Review How-To**

One skill you will learn (or practice) throughout this project is how to search and read primary literature, particularly in NLP. Please check out this pair of resources on "**How to read a paper**" and **related notes from Prof. Jason Eisner at JHU**.
For any papers you are summarizing in your proposal, you will need to describe the following (about a third of a page should be sufficient):

- What were the main objectives of the paper?
- How do the objectives relate to your project topic?
- How is what you're doing similar to what the authors of this paper did? How is it different?
- How sound do the methodologies of the paper seem to be? What are they? Are there any implicit assumptions to the model or the framework that are worth noting?
- What (briefly) were the results of the paper?
- What can you take from the paper to inform what you're doing on your final project?

## 4 Checkpoints

Each week in lab I will ask you o update me on your progress. Before the start of lab weeks with check-ins you will push an update to your **checkpoints/README.md** file. You will be graded based on your ability to:

- Make sufficient progress each week and/or be proactive about seeking assistance in the case of major roadblocks.
- Sufficiently document updated accomplishments and goals (via your **checkpoints/README.md** file in the Git repo)
- Demonstrate your progress (e.g., through code review, or analysis of results)
- Present a mid-project[4] review to the lab on December 2. This will be good practice for your final presentation.

If you need to modify the scope of your project (either to expand it or scale it down) as the semester progresses that's okay, but you'll need to discuss this with me during lab check-ins.

## 5 Presentation

Your group will present during the exam period on **December 10 from 9am-noon**. Each group will have 20 minutes to talk plus 5 minutes for questions (25 minutes maximum). The grade for this portion is completely based on your delivery, not the difficulty of the project or the

[4] an 80%-of-the-way-through project really...

impressiveness of results. Having a great project but failing to communicate your design, results, and analysis will result in a poor grade. Please work on your slides throughout the project, and be sure to practice presenting ahead of time. All members of the group are expected to present equally. Please follow Prof. Newhall's **Presentation Guidelines** for tips. Some general comments:

- Your presentation should use figures and diagrams wherever possible. In particular, you will probably have to make new figures in addition to what you plan on putting in your paper. A visual aid is always better than words on a slide.
- Slides should not be cluttered; provide a *concise* outline of main points, not a transcript of what you are going to say. You don't want the audience reading your slide, you want them paying attention to you. When in doubt, use figures and illustrations.
- **Practice....** The easiest way to handle nerves is to be comfortable with what you plan to say, and to have given a talk to an audience beforehand.
- Since talks don't normally have 4-5 people trading off speaking, make sure everyone in the group agrees who will cover which components and practice doing so ahead of time (20 minutes will go by quickly so you don't want to waste any off it agreeing how to transition between speakers).
- Know your audience: while your group has been working on your topic, nobody else in the class has. Be sure to spend an appropriate amount of time providing background motivation and describing previous work on the topic.

**Details to follow**

I will post information about the rubrics for final presentations closer to the final exam period.

## 6 Paper

The final papers should be 8 pages[5] and follow the Association for Computational Linguistics **(ACL) paper format**. If you do well and are proud of your work, you can consider submitting this to one of the *ACL[6] **Student Research Workshops (SRWs)**. If you are interested in doing this after the semester ends please get in touch with me, I am happy to help.

Your final paper is due in the **paper/** directory of your Git repository by midnight on December 15. All relevant figures and tex files should also be present there. If you use an online editor (e.g. **Overleaf**), you should still recompile your final code on the department systems to ensure compatibility.

[5] it's okay if your references spill over onto the 9th

[6] There's ACL, North-American ACL, European ACL, etc.

Note that your grade is not based on how novel your results are, but rather in your ability to convey your understanding of the problem, and how to properly frame and analyze your results. The final grade for for this is based on (a) the design and execution of the experiment (regardless of outcome) as well as (b) the thoroughness and readability of the paper.

**Details to follow**

More detailed information on the structure of final papers will be added closer to the end of the semester.

## 7 Repository Structure

**Details to follow**

(I will be requiring your repositories to follow a semi-standardized structure to keep things organized across groups. More details to follow.)

You've reached the end, great job!