Natural Language Processing (252-3005-00L): Instructions and Advice for Final Projects

Clara Meister and Ryan Cotterell

September 27, 2021

1 Introduction

If you're reading this, you have chosen to embark on the exciting endeavor that is a final research project for Introduction to Natural Language Processing (252-3005-00L). Hooray! For those of you who have never engaged in academic research before, take this project as a light, self-contained introduction to the idea. Or, as we like to call it: research with training wheels on! There are a number of steps you should take to help avoid getting stuck in frustrating situations:

- **Clearly define your goals**. At the very beginning of your project lay out exactly what you hope to achieve and create a detailed timeline and make sure everyone in your team understands them. Make sure they are reasonable goals, taking into account your current research experience, group size, and the time frame.
- **Do a thorough literature review.** In so doing, you will get a sense of which approaches are likely to fail and, of course, which approaches have already been attempted if you're trying to do something novel.
- **Consult with the course staff**. If you are feeling stuck, please email the course staff or post on Moodle. The course staff will be able to point you in the right direction.
- **Don't be scared of negative results.** As a course project, a negative result is not a failure by any means and can certainly earn full credit. We simply want to see a rigorous scientific plan laid out and executed. The findings are actually irrelevant for your grade.

Expectations

Your project should aim to demonstrate an understanding of the NLP techniques reviewed in class. You are not expected to make a scientific contribution to NLP or best a state-of-the-art method. Instead, we simply want to see a correct application of the modeling techniques you have learned applied in this course as applied to an NLP task. It is important that your summary paper provides a rigorous explanation of your experimental findings. This includes both qualitative and quantitative analysis. An example of qualitative analysis would be finding specific exemplars where your model did not work as expected. Importantly, your work will not be graded on the performance of your methods! Rather, projects will be graded holistically, taking into account criteria such as: originality, complexity of the techniques you used, thoroughness of your evaluation, amount of work put into the project, analysis quality, and write-up quality.

2 Choosing a Project Topic

Project Suitability. You can choose any topic related to NLP. In addition, your project should make substantive use of the methods taught in this class. It would not be okay, for instance, to apply a conditional random field to genomic data; nor would it be okay to build a completely rule-based NLP

system.[1] If you have any questions about what an acceptable product would entail, please feel free to contact the course staff.

Project types. Here is a non-exhaustive list of possible project types:

- tailoring an existing model, method or technique to a new application;
- transferring an existing model, method or technique from another field to NLP;
- contributing to standardization by writing a survey paper;
- proposing a new model or architecture for an NLP task;
- providing an experimental or theoretical analysis of an existing model, e.g. performing a replication study of an existing paper with ablation studies or a few more experiments on other data sets;
- proposing a new training method or evaluation scheme.

Note that encompassed in these categories is participation in a "shared task," i.e., submitting a system to one of the shared tasks proposed by a special interests group (SIG) of the Association for Computational Linguistics (ACL). Projects outside of these types are allowed but please talk to the course staff over email or on Moodle about the appropriateness of your project before the proposal deadline if your project does not fall into these categories. Our goal is not to restrict you, but to make sure you succeed!

Example Projects. Other educational institutions that offer NLP courses sometimes have a similarly structured course project, e.g., Stanford's CS 224n.[2] You can look at their sample projects for inspiration and to get a sense of what will be expected.

2.1 Literature Review

You will be expected to perform a thorough literature review such that you can both summarize and cite relevant work in your write-up. We recommend doing this sooner rather than later as it is generally much easier to define your project if there is existing published research on a similar task or modeling technique. Identifying existing relevant research—including published code—will ultimately save you time and make the process more enjoyable. Prior works can give a sense of best practices for approaching projects, vet out initial ideas, and lay the groundwork for your own research. Finding relevant research is not always easy. However, there are a number of places to start looking:

- Recent publications at any of the top venues where NLP research is published: e.g., ACL, EMNLP, TACL, NAACL, EACL. All of these are available in the ACL anthology:
- Use a keyword search in the following search engines/repositories:
  - http://scholar.google.com
  - http://dl.acm.org/
  - http://arxiv.org/
  - https://search.library.ethz.ch/
- Look at publications from NLP research groups. For example:
  - https://rycolab.github.io/publication
  - https://www.clsp.jhu.edu/publications/
  - https://copenlu.github.io/publication/
  - https://nlp.stanford.edu/pubs/
  - https://nlp.cs.ucl.ac.uk/#/publications
  - https://www.cs.washington.edu/research/nlp/publications-by-year

[1] This is not a judgement about the usefulness of rule-based NLP. Rather, we simply require that you demonstrate mastery of the techniques that are the subject of this class.

[2] http://web.stanford.edu/class/cs224n/project.html

2.2 Finding Data

Here we present options for finding natural language data for your project.

Shared Task datasets. Shared tasks (also called bake-offs) are competitions to which researchers submit systems that address specific, predefined challenges. Often, the data provided by the organizers of these tasks is well documented, clean, and standardized. This often makes these datasets incredibly user-friendly. Additionally, the shared task websites will also frequently provide baseline systems, which may be useful for your own comparison. You can find a (quite non-exhaustive) list of shared tasks at the following link, but we recommend looking directly at websites of special interests groups: https://www.aclweb.org/portal/category/topics/shared-task. One shared task currently running that you could participate in is on numbers in financial texts: https://sites.google.com/nlg.csie.ntu.edu.tw/finnum3/.

Using publicly available datasets. There are a number of online resources that host standard NLP datasets. We present a few here:

- NLP Progress and Papers with Code: Repositories that track state-of-the-art performance for the most common NLP tasks. They also includes many of the common datasets for these tasks.
- Social Media and Online Platform Data: many online platforms provide APIs or readily compiled data dump files. Consider Twitter, Reddit, YouTube transcripts, Wikipedia, Reuters.
- Kaggle: a platform for hosting competitions on tasks that also provides many datasets.
- Alphabetical list of datasets in the public domain with text data for use in Natural Language Processing: https://github.com/niderhoff/nlp-datasets.
- Emailing the Authors: If you come across an interesting dataset that is not publicly available, you may consider contacting the authors. Often, authors of academic papers are more than happy to email you a copy of their dataset.

Just because a dataset is online and already in a single, structured file does not mean the dataset has been processed in a manner suitable for your project. Take care to analyze any online resources to first, ensure their quality and second, determine what additional steps you may need to take in order to properly format the data.

Collecting new data. While it is possible to collect your own data for your project, we do not recommend it. Data collection is often a time-consuming and messy process that is more difficult than it appears. If you choose to do so, make sure to budget the data collection time into your project and provide a thorough explanation of your processing techniques in the write-up. Note that your project must have a substantial modeling component; if you spend all your time on data collection and none applying modeling techniques, your grade will suffer.

3 Advice

3.1 Data Processing

Data processing, even for already "clean" datasets, can be cumbersome. It is worth investing time in a thoughtful, well constructed pipeline to assure you do not have to redo experiments. The following should all be considered:

Cleaning. You should consider if your data need to be tokenized, parsed, or annotated. The following widely used and well documented resources may be helpful:

- NLTK: a lightweight NLP toolkit written in Python;
- Moses: an SMT library that provides tokenization and cleaning scripts with coverage in many languages.
- StanfordNLP: a Python library providing tokenization, tagging, parsing, and other capabilities.

Splitting. At the beginning of your project, you should split your data set into training data (most of your data), validation data, and test data. A typical train–dev–test split might be 80–10–10. Many NLP datasets come with predefined splits, and, if you want to compare against existing work on the same dataset, you should use the same splits. Explicitly, **training data** can be used to estimate the parameters of your model. The **validation data** should be used for model selection, but not parameter estimation. In terms of neural modeling, model selection may consist of choosing the best hyperparameters for an architecture or deciding whether to stop your optimizer early to avoid overfitting. The **test data** should only play a role at the end of your project. You should only evaluate your model *once* on the test data. Looking at the test data repeatedly is knowing as p-hacking or data dredging.

Preliminary analysis. Before modeling the data, performing a preliminary analysis of the data is helpful for successfully completing the project. Actually looking at your data may reveal flaws, e.g., missing entries, inconsistencies, or oddities, e.g., unexpected trends, noise. When researchers fail to look at the data, they often discover such inconsistencies at a later and much more inconvenient time. Some recommended approaches include building visualizations and computing basic statistics. For example, does your text data follow an expected Zipfian distribution? What is the variance of various attributes, such as length or score (if applicable)? While we do not require such analysis in your writeup, we nonetheless recommend it. There are a number of online resources[3] that provide guidance on how to do such an analysis.

3.2 Baselines

NLP is largely an empirical science in that most recent progress in the field was made through experimentation. For this reason, good experimental practice is incredibly important when doing NLP research. One aspect of good experimental practice is selecting a good baseline to benchmark your method and properly comparing against that method. Without a fair comparison to previously proposed methods, we cannot tell whether an innovation is actually an improvement over what the field currently offers. For instance, comparing against a simple baseline may reveal that your task does not actually require a complex, resource-intensive model. In general, simpler methods should be preferred.[4] Regardless of your chosen research topic, we will expect some sort of baseline comparison in your final report. Note that some baselines are theoretical. Baselines can include simpler models or models proposed by other works. You should first make an effort to understand what the baselines are. For example, suppose you're building a multilayer LSTM-based network to do binary sentiment analysis. The simplest baseline is the guessing baseline, which would achieve 50% accuracy (assuming the dataset is 50% positive and 50% negative), or the mode baseline, which would achieve the approximately the accuracy of the positive/negative split in the training set. A more complex baseline would be a simple non-neural algorithm, such as logistic regression or a Naive Bayes classifier. Alternatively, you may choose to reimplement a baseline from another work. Often, researchers will open-source their code so look online before starting the grueling process of building a complex model from scratch!

3.3 Use of Existing Infrastructure

There are a number of software tools and packages that can aid in your research. You are allowed (and encouraged) to make use of these tools, however, if you do so, please be explicit about it in your writeup. These include:

- PyTorch `https://pytorch.org/`
- TensorFlow `https://www.tensorflow.org/`
- scikit-learn `https://scikit-learn.org/stable/`
- HuggingFace Transformers `https://github.com/huggingface/transformers`

[3] such as `http://seas3.elte.hu/lingtheo/notes/gkiss-data-in-lcs.html` or `https://r4ds.had.co.nz/`

[4] Occam's razor is good maxim to be aware of when doing research.

- fairseq https://fairseq.readthedocs.io/en/latest/
- Various other packages in R, Python, Julia, etc.

As is the case with collecting data from scratch, we do not recommend building complex neural models from scratch. While coding an LSTM from scratch may be an interesting exercise, it will likely take more time than you want to spend. After all, the goal of this project is to answer a research question. Note that training a predefined model in a deep learning library with readily available preprocessed data (e.g., reproducing one of the task README's in fairseq) does not constitute a full project! However, making use of pre-trained models in your modeling process is fine. If you are unsure of your practices, please contact the course staff.

3.4 Computing Resources

We can give groups access to the Euler HPC cluster for their projects, which provides limited access to GPUs. While much of today's NLP research makes heavy use of computational resources, many interesting analyses can still be done without utilizing GPUs for days on end! As such, please note that you are not required (nor encouraged) to train new, large neural networks or do extensive hyperparameter searches as this may not be feasible given the resources that we can provide.

4 Deliverables

All deliverables will contribute to your final project grade. After submission of proposals, each group will be assigned to one of the TAs. If any questions arise during your project, you may schedule a meeting with your assigned TA in between each of the deadlines to discuss them. Please reach out to your assigned TA with ample time as they likely have scheduling constraints. **Submissions:** There will be a centralized submission process via Moodle for the project proposal; only one member from each group should submit. After this, all deliverables should be sent directly to your assigned TA.

- **Proposal** (5%) *Due Date: 31/10/2021*

  Your initial plan should be 1–2 pages in length (not strictly) and should, at a high level, provide answers to the following questions:

  - What are the specific details of the task you are investigating? What is the input and what is the output? Give concrete examples. If the task can't be framed as input and output, what exactly are you trying to achieve?
  - What dataset(s) and tools are you planning to use?
  - What does success look like for your project?

  Please be sure to include all group members' names and nethz usernames in the proposal. Further, if you need access to computational resources (i.e., the Euler HPC cluster), please specify this in the header of your proposal.

- **Progress Report** (5%) *Due Date: 15/12/2021*

  The progress report should detail what you have accomplished so far and what you are planning for the future. There are no length expectations or constraints, but the progress report should include, at the least, a draft of the introduction, a literature review that comments on related work, and some preliminary results (e.g., baseline system performance and dataset statistics). Furthermore, any changes from your initial plan should be noted. **The point of the progress report is to ensure that you get a high grade. The teaching staff hopes to spot any issues and resolve them before they grade the final write-up.**

- **Final Write-up** (60%) *Due Date: 15/1/2022*

  The final report should take the form of a research report. You write-up should, at a minimum, include an introduction, background section, theoretical explanation of modeling techniques,

experimental findings, and analysis. You are **required** to use ACL.[5] We recommend looking at the structure of other NLP research papers as a guideline for your write-up's layout.

We additionally expect you to submit any code used during your experiments, along with a README giving instructions on how to reproduce your results. Reproducibility is a fundamental part of research afterall. The simplest way of doing this is to create a GitHub repository. For the teaching staff, it would be easiest if you give access to the "rycolab" GitHub organization). However, a tarball with your code is also acceptable.

- **Presentation** (30%) *Due Date: 18/1/2022*

  Each group must give an 15-minute presentation on their project.[6] While there are items we want you to cover, the exact format is up to you. You may, for example, give a live demo or create a series of infographics. Regardless of the format you choose, note that we are expecting that the presentations provide appropriate background information for the project and discussion of final results of the project. Due to the COVID-19 situation, we ask that you **pre-record them** and send the video file directly to your assigned TA.

[5] Please uncomment the "final copy" command.

[6] While we do not restrict you to 15 minutes, all critical information must be conveyed in the first 15 minutes as the TA team may not have time to watch $>$15 minutes of each presentation.