NLP Final Project Guidelines

Objectives and background

The purpose of the final project is two-fold. First, it will give you the opportunity and incentive to study in-depth a topic that interests you. Second, it will test your ability to use NLP algorithms to solve a problem within this topic. The project definition is purposefully open-ended. The goal is for you to be able to spend time thinking deeply about NLP and how to best apply it in a real-life scenario.

Final projects are to be done in teams of 1, 2, or 3. Your final deliverables include a proposal, final presentation, final paper, and final source code (e.g. runnable Colab notebook or Python files with README showing how to run it). To give you an idea of the scope, we are expecting you to spend ~40 hours (per person) between now and the end of the semester on the project.

Project proposal

A 1-2 page PDF of your project proposal is *due by 11:59pm Apr 11 via Gradescope*. We will send you feedback ~1 week later. Please note that your project proposal will factor into your overall project grade, so make sure that it is written well and follows the guidelines provided below.

You are free to choose any topic that you are excited about, try to optimize for the world and then reduce it to something realistic. You must make sure that it's feasible (please check with us when in doubt). Your project proposal must detail the data that you plan to use, how you will pre-process it, the questions you will ask, the NLP algorithms you expect to use (permitted to change), and what you expect to get out of your project. Most importantly, you should demonstrate in the proposal that you have gained familiarity with the data set.

Below are just some examples about what you can do:

1. Frequently asked question answering for Covid-19
2. Predicting the most highlighted sentences in Chinese novels.
3. Chinese poetry generation based on user-specific keywords
4. A food ordering bot for NYUSH cafeteria
5. Machine translation for under-resourced languages

Final presentation

The final presentation is an opportunity for you to showcase your work. You will probably want to follow a structure similar to:

1. Problem situation: what is the problem? Reflect on why it's important. What are the difficulties?
2. Descriptive analytics: Which data did you get? Any interesting insights/visualizations?

3. Machine Learning: How did you tackle the problem?
4. Future work: If you had infinite time, what would you envision the next steps to take are? How will you improve the results?

Final paper

Your final paper should read like a mini conference paper. It should contain the following sections:

- Abstract: a short description of the whole paper, highlighting important results.
- Introduction: introduction to the problem, potentially relating to prior work done by other people.
- The dataset and features: a concise explanation of the dataset and features, where did you get it from? How did you process the data? Did you combine multiple datasets, and if so, how?
- Explanation of the method used: don't explain the actual algorithm (do mention it), but rather how you used the algorithm to solve the problem. If you used a special loss function or deviated in any other way from the standard algorithm, this is where you mention it.
- Results: walk the reader through your main results, how well did your model perform? Highlight both positive results and shortcomings of your approach.
- Future work/Conclusion: reflect on your findings and think of how you would solve this problem given that you would be able to work on it for 5 more years. Which data would you try getting? Which method would you use? How would it help the problem?

Less is more, but you will probably want to reach about 4-5 pages. We recommend you to work with LaTeX to write the document, but you are free to choose whichever editor you like. You may use https://www.overleaf.com/ to edit latex documents online in a collaborative manner. Conference paper templates for ACL can be accessed here: https://www.overleaf.com/latex/templates/instructions-for-acl-ijcnlp-2021-proceedings/mhxffkjdwymb

Final implementation

Your project implementation should be done in Pytorch or Tensorflow. Your implementation should demonstrate competency on the following aspects:

- Demonstrate the ability to implement the full deep NLP workflow: Data preprocessing & tokenization of project data; neural modeling; model training & tuning; evaluation on unseen data which involves prediction/decoding
- You may use the basic neural network components (e.g. LSTM, GRU, Transformers, BERT etc). But you cannot use high-level black box (e.g. Pipelines) that does everything for you. This is not good for the purpose of learning.
- Implement a reasonably good baseline.
- Choose reasonable evaluation metrics and report evaluation results.
- Perform error analysis: Figure out error patterns and describe what you found.
- Propose one technique to improve the baseline based on the error analysis. For example, you may use feature engineering to enrich your input features. Tuning hyperparameters such as learning rates, batch size, beam size etc are not considered as technique.

- Demo your output. (e.g. In an FAQ bot, you input a query to your model, and your model outputs top related answers with relevant scores; For Chinese poetry generation, your model outputs a poem with likelihood scores etc)

Resources-data

Below are some suggested ways to obtain data, depending on the need of your project:

- Huggingface datasets library (covers hundreds of public sharable corpus): https://huggingface.co/datasets
- Linguistic Data Consortium (LDC): NYU has an account. Professor can help request the data. https://catalog.ldc.upenn.edu/
- Kaggle
- Google search
- Crawl your own data from the Web.
- WMT data corpora for machine translation (Search keyword "WMT machine translation workshop" in Google"
- Movie subtitles in multiple languages: https://www.opensubtitles.org/en/search/subs
- Chinese poetry: https://github.com/chinese-poetry/chinese-poetry

Pretrained models (BERT and its variants):

- Pretrained neural model repository for finetuning: https://huggingface.co/models

Resources-cloud computing

Since I would expect you to apply deep learning for your project, you may want to use Google Cloud / Colab, Microsoft Azure or Amazon AWS. Each of these has a free track that allows you to perform computations up to a certain point. For computationally heavy projects and needs to run training for few days, we can use NYU Shanghai's HPC. Application form is here: https://shanghai.nyu.edu/it/hpc and professor can sponsor.

If you are willing to spend some money, you can upgrade Colab into Colab Pro for $9.99 USD per month. It gives you longer running notebooks, less timeout, and a faster GPU etc. You can cancel the upgrade after you finish the project. https://colab.research.google.com/signup#

Grading Rubric

Component (Weight, total = 100 points)	**Grade**
Proposal (20 points)	
The proposal clearly specifies the problem settings and shows that you have done some preliminary research on this problem already. It motivates why you think it's important and what the role of machine learning will be to solve the problem. The data description includes at least one relevant dataset and potentially other sources where you think you may be able to get data.	A to B

The proposal is vague and unclear for those who have not been a priori exposed to it. It may be ill-motivated, contain grammar or spelling errors and the proposed data sources are not realistically obtainable or not well-connected to the problem.	B- to F
Implementation (30 points)	
The code is neatly written and solves the problem correctly. The logical flow of your project notebook makes sense and starts with an exploratory analysis. You correctly attribute other people's source code using references and comments when needed. Higher grades will be awarded to code that is well-documented, concise, vectorized (avoids for loops) and clear. A top grade is reserved for notebooks that are, in essence, flawless on all dimensions.	A to B+
The code works but contains some problems which may vary in severity. It is not easily readable by other people because of a lack of good documentation/comments. There may be issues to the flow of the code execution or parts may be missing. You may also not have used vectorization to its fullest effect or left out other potential improvements to optimize your code in terms of speed or accuracy.	B to C
Grades in this range are reserved for notebooks containing fundamental problems in the analysis (such as confusing causation with correlation, or not using a test-set). The code may not run or is solving a different problem. Acts of plagiarism, cheating are considered serious fundamental flaws.	C- to F
Final paper (30 points)	
The final paper is appropriate for the intended audience and covers all important aspects of the analysis. Information regarding the setting is of outstanding quality (recent, useful, relevant) and based on well-research and competently used external sources which are properly attributed to their original sources in (Author, Date) format when needed. Higher grades will be awarded to documents that are concise, well-written, use LaTeX, are persuasive and clear. A top grade is reserved for documents that are, in essence, flawless on all dimensions.	A to B+
Grades in this range will be awarded when the final paper leaves out important ideas, or when it demonstrates a lack of familiarity with the underlying problem or previous solutions to the problem. It may also be logically inconsistent, or problematic in any other way. Especially poorly written documents might earn grades in this range, particularly if the core problem and solution are not well connected or have not been clearly communicated.	B to C

Grades in this range will be awarded if there are serious issues with the final paper, including but not limited to acts of plagiarism, a failure to take into account concepts taught in class (such as making basic errors), or writing quality that impedes understanding.	C- to F

Final presentation (20 points)

The type of presentation is appropriate for the topic and audience, the introduction is attention-grabbing, lays out the problem well, and establishes a framework for the rest of the presentation. The information is presented in a logical sequence. The solution you came up with is logical, relevant and intuitively follows from the rest of the information. The speaker maintains good eye-contact, is appropriately animated and communicates information in a clear, controlled and smooth manner. The length of the presentation is within the assigned time limits. A top grade is reserved for presentations that are, in essence, flawless on all dimensions.	A to B+
The presentation contains all the relevant information, but may be hard to follow because it is disorganized, too many texts in a slide, or not communicated clearly by the speaker. The content may have been presented in an uninspired manner and, the speaker may not have been well prepared (or is holding flashcards). Furthermore, there may be issues with any of the following, which impede clarity: visual aids are hard to read or otherwise problematic (label your axes!), the speaker does not come across as knowledgeable or professional.	B to C
There are serious issues with the presentation which impeded comprehension of the topic: unfamiliarity with the problem setting, not being able to answer basic questions about your solution, acts of plagiarism, ...	C- to F