On one hand, machine learning is such an exciting discipline because of its rapid pace of innovation, with new technologies and capabilities emerging at every turn. On the other hand, mastery of machine learning is so challenging because it evolves so rapidly that any frozen skill set (however masterful) risks becoming outdated. In such a dynamic environment, what skills will stand the test of time and continue to serve you, long after many of the precise methods taught in this class go out of fashion?

Whether you plan to become a builder in industry or a scholar in academia, perhaps the most important meta-skill to acquire is to be able to think like a researcher. Thinking like a researcher means (i) accessing new knowledge from a messy scientific literature, pulling out the useful and correct bits from papers while discarding (or correcting) mistakes; (ii) applying new methods and ideas to real problems that you are trying to solve.

For the course project, we offer two options, (i) a “researcher literature”, focused on performing a comprehensive review of a research subfield; and (ii) a builder track, more focused on applying methods (both covered in the course and not) to a real problem, with the challenge of producing useful results and a comprehensive report. In both cases, you will be expected to read, digest, and implement, but the emphasis is different. The *main dish* of the “research literature” track is a comprehensive report synthesizing knowledge from the field, with experiments performed for didactic purposes, to demonstrate understanding and reproducibility. The main dish of the “builder track” is a working system for a specified task, with the report synthesizing related work and providing a comprehensive write-up of all techniques that were tried and the results that they achieved.

Regardless of which track you choose to pursue, the following policies will apply:

1. You must work on the course project in groups of 3 or 4; **you may not work on the course project alone.**

2. There are three deliverables associated with the project; the contents of each deliverable will depend on your choice of track (see the track specific details below):

    a. **A 2-page proposal, due November 3rd at 11:59 PM** - these will largely be graded on completion although you may be asked to resubmit unsatisfactory work; their primary purpose is to catch and correct any early misconceptions about the project task/scope. The proposal is worth 3% of your final grade.

    b. **A 4-page check-in, due November 17th at 11:59 PM** - these will also largely be graded on completion and serve as a way for us to ensure that you’re making sufficient progress on the project. The check-in is worth 3% of your final grade.

    c. **An 8-page final report, due December 8th at 11:59 PM** - the final report is worth 20% of your final grade.

3. The page limits for the deliverables defined above are strict, with the exception of pages only containing references/citations, which will not count against this upper limit.

4. Each group will be assigned a course staff member as their project mentor: they will be your primary point of contact for any questions that arise during the course of the project.

    a. After submitting your proposal, your group must schedule a meeting with your project mentor where you will collectively review your proposal.

    b. Based on this meeting, your group will either receive approval to move forward or be asked to resubmit your proposal, incorporating feedback from the meeting.

Research Track

The final deliverable of the research literature track will be an 8 page report focused on an emerging area in machine learning. Groups are expected to read deeply into their chosen sub-area and the final report should paint a comprehensive picture of both the current state of the art, its connections to related areas of research, and its historical context (i.e., connections to ideas in the research literature that preceded it). The chosen areas are all hotbeds of activity and part of the challenge of the project is determining which recent papers are worth focusing on and why even before the final word has been settled.

Suggested topics include the following: (other topics may be considered with instructor approval)

- Incorporating human preferences into language models [and/or] image generation models.
- Compression of language models
- Efficient fine-tuning of language models
- Ensuring / assessing factuality in machine-generated text
- Memorization in neural networks
- Integrating causal modeling and representation learning
- Robustness of machine learning models against adversarially perturbed input.
- Privacy-preserving deep learning
- Assessing discrimination/bias in [LLMs or image-generation models].

Guidelines for the report:

- Typeset the report in LaTeX (you can use Overleaf).
- The report should be 8 pages, single spaced.
- Use a generic manuscript skin (Not a NeurIPS paper template)

- Each report should contain an abstract (one solid paragraph), an introduction (1-2 solid pages) and then a clearly organized exposition that provides a useful framework for thinking about works in the literature.

Builder Track

If you select this option, you will explore different ways of performing machine translation, a common natural language processing task. You will use this Kaggle dataset, consisting of English sentences and their French translations; **we highly recommend that you read sections 1 through 3 of the associated paper** to familiarize yourselves with the dataset. Your goal is to use this dataset to train machine learning models that take English sentences as inputs and return the same sentence in French.

You will solve this task using three approaches:

1. From scratch - first, you will build and train *three* distinct methods for performing machine translation: these should be coded entirely by your group, with no reliance on existing models, large or otherwise. That being said, you are free to use any Python packages you wish including PyTorch. For the purposes of this section, methods that correspond to the same underlying model just with different hyperparameters (e.g., feed-forward neural networks with differing numbers of hidden layers) do not count as distinct. However, different classes of neural architectures (e.g., RNNs vs. feed-forward neural networks) are considered distinct; your project mentor will be able to provide feedback on this aspect of your project. At least one of your methods must include some component (e.g., an architecture, embedding, optimizer, etc…) first published in a *contemporary* research paper on machine translation i.e., something published in a top machine learning conference in the past 3 years.

2. Relying solely on existing LLMs - under this approach, you will use a previously trained LLM, specifically Meta's Llama-2-7B model, which you can download here. You should read through the instructions for setting up and interfacing with Llama 2 as well as the technical specifications. You may use either the pretrained `text-completion` version of the model or the fine-tuned `chat-completion` version. Your code for this approach should only consist of calls to the Llama-2-7B API; **you may not retrain or fine-tune the model for the task or update the parameters/hyperparameters in any way.**

3. Building on existing models - for this final approach, you should continue working with the Llama-2-7B model but you must adjust, retrain/fine-tune, or extend the pretrained model in some way. How you do so is completely up to you! We encourage you to get creative in this section and run lots of experiments, maybe do some independent research, with the hope of improving upon your results from the previous two sections.

Here are a few more general specifications that will apply to each of the approaches above:

- All of your implementations must be completed in Python; at the end of the project, you will submit all the code you wrote along with your final report so please do follow good coding practices and document your work.

- To standardize the comparison across methods and groups, you must reserve the last 10% of the dataset as a test dataset and report each methods' performance using this dataset. You are free to partition the remaining 90% of the dataset however you wish.

- You will evaluate each of your models using the BLEU metric; you may use pre-built methods to compute this score for you e.g., PyTorch's Torchtext implementation.

Deliverable specifications:

1. Your proposal must contain at minimum:

    a. a description of the three methods you intend to implement from scratch and

    b. screenshots from *at least two* group members demonstrating that they were able to download Llama-2-7B (both versions) and successfully run the `example_chat_completion.py` script.

    Of course, you are free to include any additional plans/ideas you have regarding approaches 2 and 3 at that point in time.

2. Your check-in must contain at minimum:

    a. Initial results for all of your from scratch implementations (of course, these do not need to be the final product; we encourage you to keep iterating on these until the final report deadline!) and

    b. a description of how you plan to modify the Llama-2-7B model for approach 3.

3. Details about the final report will be released closer to the deadline but at a high-level, the final report should include the relevant details of all the methods you implemented, the major findings from each approach, and some analysis of the results

Nous vous souhaitons le meilleur de la chance; impressionnez-nous!

Final Report Specifications

Regardless of which track your group chose to pursue, your final report will consist of three deliverables. There will be a separate Gradescope submission for each of the following items:

1. A writeup: the details for what should go into the writeup will differ between the two tracks (see below). However, regardless of which track your group chose, the following shared requirements must be met:

    a. The writeup can be at most 8 pages, single-spaced. Pages containing only references/citations do not count against this upper limit.

    b. You must typeset the writeup in LaTeX using the provided template, found here: https://www.overleaf.com/read/fmjqyvpybvjd#237105

    c. The writeup must have a descriptive title and contain the names and AndrewIDs of all group members who at some point contributed to the project, regardless of whether or not they are still enrolled in the course.

    d. Submit the writeup as a group in Gradescope; you should have one submission per group.

2. A statement of individual work: each group member must independently write a short paragraph describing their contributions to the project and submit them individually to Gradescope. These will not be graded and will only be referenced in the (unlikely) event that we need to assign different grades to separate group members.

3. All code written for the project: code will be assessed differently for the two tracks. However, every group must submit all the code they wrote to Gradescope. You may submit as many files as you need. Each file must have a meaningful name so that your project mentor can easily identify its purpose. If you wrote code in the form of Python notebooks, please convert those to .py files before submitting them to Gradescope. Submit your code as a group; you should have one code submission per group.

Builder Track Specific Details

Your writeup must at minimum contain the following components:

1. Title and Author List

2. Problem and Dataset: Briefly describe the task and motivate its importance. Then describe the dataset, including where the data is sourced from and any potential limitations, issues or biases the data might suffer from.

3. Methods: The bulk of your writeup should be a thorough, detailed description of all the models your group implemented for each of the approaches. Crucially, you must demonstrate a deep understanding of all the methods you implemented, including your contemporary method(s); **simply providing a list or screenshots of your code is insufficient**. In addition, you should describe the training procedure(s) and hyperparameter optimization techniques. From your report, a technical reader should be able to replicate your results by following these descriptions, i.e., there should be no ambiguity as to how you implement your model. If applicable, you should also briefly detail any approaches you tried but ended up not working well.

4. Experimental Results: Show plots and/or tables of the performance of your algorithms and interpret what they mean; be sure to label all of your figures and explain the findings. You must also define all performance metrics you used for evaluation. Describe how the results in each of the experiments aligned or didn't align with your expectations. Make sure to provide confidence intervals where appropriate or standard errors when comparing methods.

5. Discussion and Analysis: Finally, analyze your models and their corresponding results. Provide explanations for the relative performances you observed and highlight any limitations/shortcomings of your approaches. Comment on how you would further improve your methods.

6. References

The following rubric will be used to assess the writeups for builder track projects:

- Completeness (20 pts) - all of the required components are present in your project i.e., three from-scratch methods with at least one contemporary component, an exploration of in-context learning with some existing LLM and a fine-tuning based approach.

- Technical Soundness (30 pts) - the methods you implement must be described in sufficient technical detail such that your project mentor can properly assess your work; a good rule of thumb is that a well-informed practitioner should be able to recreate your methods entirely from their descriptions. Your methods should follow rational machine learning principles/best practices as covered in the course e.g., hyperparameter tuning should be done with a held-out validation dataset and not on the test dataset.

- Implementation Correctness (30 pts) - your code matches the description of your methods in the writeup; for this portion of your grade, we will manually inspect all the code you submit. As such, your group should follow good coding practices, e.g., meaningful variable names and detailed comments. We reserve the right to deduct points if your code is unintelligible (given a good faith effort by your project mentor).

- Clarity (10 pts) - this portion of your grade will assess the quality and organization of your writeup; it is crucial that you present your work in a clear and understandable way.

- Formatting (5 pts) - your writeup must adhere to the guidelines we have established above e.g., it respects the 8 page limit and uses the correct LaTeX template (https://www.overleaf.com/read/fmjqyvpybvjd#237105)

- Performance (5 pts) - finally, a small portion of your group's grade will be based on how well your methods work. Crucially, this is not a cross-group competition: many groups are exploring fundamentally different methods that render comparisons meaningless. Any set of reasonable BLEU scores will receive the majority of the credit for this rubric item, with a small portion being reserved for truly exceptional performance.

Researcher Track Specific Details

Your writeup must at minimum contain the following components:

1. Title and Author List

2. Topic: Briefly describe the focus of your literature review and motivate its importance. For the purposes of completeness, you should not only describe the topic but also any related terms/fields that your group deemed out of scope for your review.

3. Literature Review: The bulk of your writeup will be a thorough review of the academic landscape surrounding your chosen topic. It should begin with the historical context: what were some of the pioneering works in the field and how did they influence more recent research? When discussing the current state of research in your chosen topic, **it is crucial that you do more than just list papers and methods**: you should analyze the content of the works that you've read by e.g., drawing connections between different lines of inquiry, comparing and contrasting approaches, finding limitations or weaknesses in one paper that are addressed by another, etc…

4. Experimental Results: Briefly describe what method(s) your group implemented and the empirical settings in which you evaluated your implementation. Show plots and/or tables of the performance of your method(s) and interpret what they mean; be sure to label all of your figures and explain the findings. You must also define any performance metrics you used for evaluation.

5. Discussion and Analysis: Finally, you should reflect on the overall state of your chosen topic based off of the review your group performed. This could include (but is not limited to) ideas about where the field as a whole is moving towards, what the promising new avenues of research are and conversely which methods do not show a lot of promise or are likely to be subsumed by alternatives, etc… These should not be entirely speculative but should be grounded in your understanding of the state of research in your chosen

topic; of course, some (informed) imagination is encouraged here!

6. References

The following rubric will be used to assess the write-ups for researcher track projects:

- Thoroughness (20 pts) - your review should cover all the seminal works and major sub-areas within your chosen topic, as deemed appropriate by your project mentor. That being said, you should not attempt to include all papers that are even remotely related to your chosen topic; rather, an important skill that will be assessed here is how well your group can identify the influential works in a field.

- Historical Contextualization (15 pts) - as we have been doing in lecture this semester, it can be insightful to ground the current state of research using the history of the field. Your literature review should go as far back as possible, trace the development of your chosen topic from the earliest related references you can find and draw connections to contemporary work.

- Analytical Depth (30 pts) - the most important part of any literature review is synthesis i.e., drawing connections and identifying trends in the area. The bulk of your writeup grade will be determined by the level of analysis your group performs when reviewing the research. A simple list of papers is not sufficient: your writeup must demonstrate a deeper understanding of the research area.

- Empirical Quality (20 pts) - the method(s) and experiment(s) you implement must be described in sufficient technical detail such that your project mentor can properly assess your work. Your method(s) should also follow rational machine learning principles/best practices as covered in the course e.g., hyperparameter tuning should be done with a held-out validation dataset and not on the test dataset. A portion of this rubric item will also be how well your submitted code matches your description in the writeup. You should also motivate your experiment(s): what is the connection to the literature review and what insights are you trying to highlight?

- Clarity (10 pts) - this portion of your grade will assess the quality and organization of your writeup; it is crucial that you present your work in a clear and understandable way.

- Formatting (5 pts) - your writeup must adhere to the guidelines we have established above e.g., it respects the 8 page limit and uses the correct LaTeX template (https://www.overleaf.com/read/fmjqyvpybvjd#237105)