Project: What to do?

- Default project: **BERT** + **Fine-tuning on downstream tasks**
- Examples:
- BERT + Sequence Classification
  - Sentiment classification
  - Paraphrase detection
  - Semantic similarity etc.

Code template provided

- Or, BERT + QA on SQuAD, TriviaQA, Natural Questions etc.
- Or, BERT + Translation

BERT + Sequence Classification

Source code credit to: Stanford CS 224N Winter 2023 Default Final Project: Multitask BERT https://github.com/gpoesia/minbert-default-final-project

- Primary Task: **Sentiment classification**
- Training dataset: Stanford Sentiment Treebank (SST) on movies
  - Train: 8545 lines of (sentence, score) pairs; score from 1 (neg) to 4 (pos)
  - Dev: 1102 lines
- Requirement
  - Finish the implementation of BERT (bert.py, skeleton provided, with six TODOs); Initialized from pretrained model
  - Fine-tune it on SST data (classifier.py, mostly implemented with two TODOs)
  - Extend and improve it in various ways:
    - Multi-task task through **paraphrase detection** and **semantic similarity regression** tasks (multitask_classifier.py, three new TODOs)
    - Different tasks correspond to different predict_xxx() functions in forward function

What to do with custom projects

- If you:
    - Have some research project that you’re excited about (and are possibly already working on)
    - You want to try to do something different
    - You want to see more of the process of defining a research goal, finding data and tools, and working out something you could do that is interesting, and how to evaluate it
- Then: Do the custom final project
- **Requirement**: must substantively involves both human language and neural networks

Project: What not to do?

- Train BIG models from scratch
  - Be realistic about the scale of compute you can do
  - You do not have the resources to train your own GPT-2 model from scratch
  - You probability do not have the resources to load a 7- to 11-B model (Llama-2, ChatGLM-3, Mistral-7B, T5-11B etc.)

Some trending topics

- Evaluating and improving models for something other than accuracy
  - Adaptation when there is domain shift
  - Evaluating the robustness of models in general
- Empirical work looking at what large pre-trained models have learned
- Get knowledge and good task performance without much data
- Bias, trustworthiness, and interpretability of large models
- Low resource languages or problems

Some trending topics

- Building small, performant models can be cool!
- Model pruning/quantization
  - QLoRA; Pruning; Compression: https://proceedings.mlr.press/v119/li20m/li20m.pdf; https://arxiv.org/pdf/2004.07320
  - Efficient Open-domain QA: https://efficientqa.github.io/ (within 6GB mem)
- Baby LM challenge: https://babylm.github.io/index.html
  - Efforts on optimizing pretraining given data limitations inspired by human development
  - 100M to 10M word text data

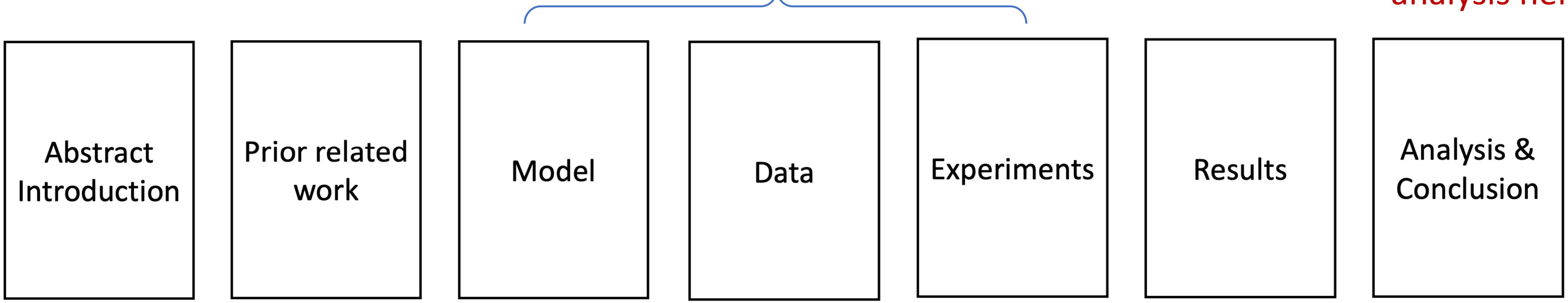
Grading: Project and Presentation

- Write-up quality

Try creative combos of model + data + method

It’s okay your acc is low!

But do include analysis here!



Abstract Introduction

Prior related work

Model

Data

Experiments

Results

Analysis & Conclusion

- Focus on what you have done
  -- not on the amazing ChatGPT output showing that “look, it works zero-shot”
- Minimal 5 pages (template provided)

Important Dates

- In-class presentation of project: Week 16, Tuesday, June 4th, 2024
  - 7 minutes presentation + 3 minutes QA
- Project report due: Friday 11:59 PM, June 7th, 2024