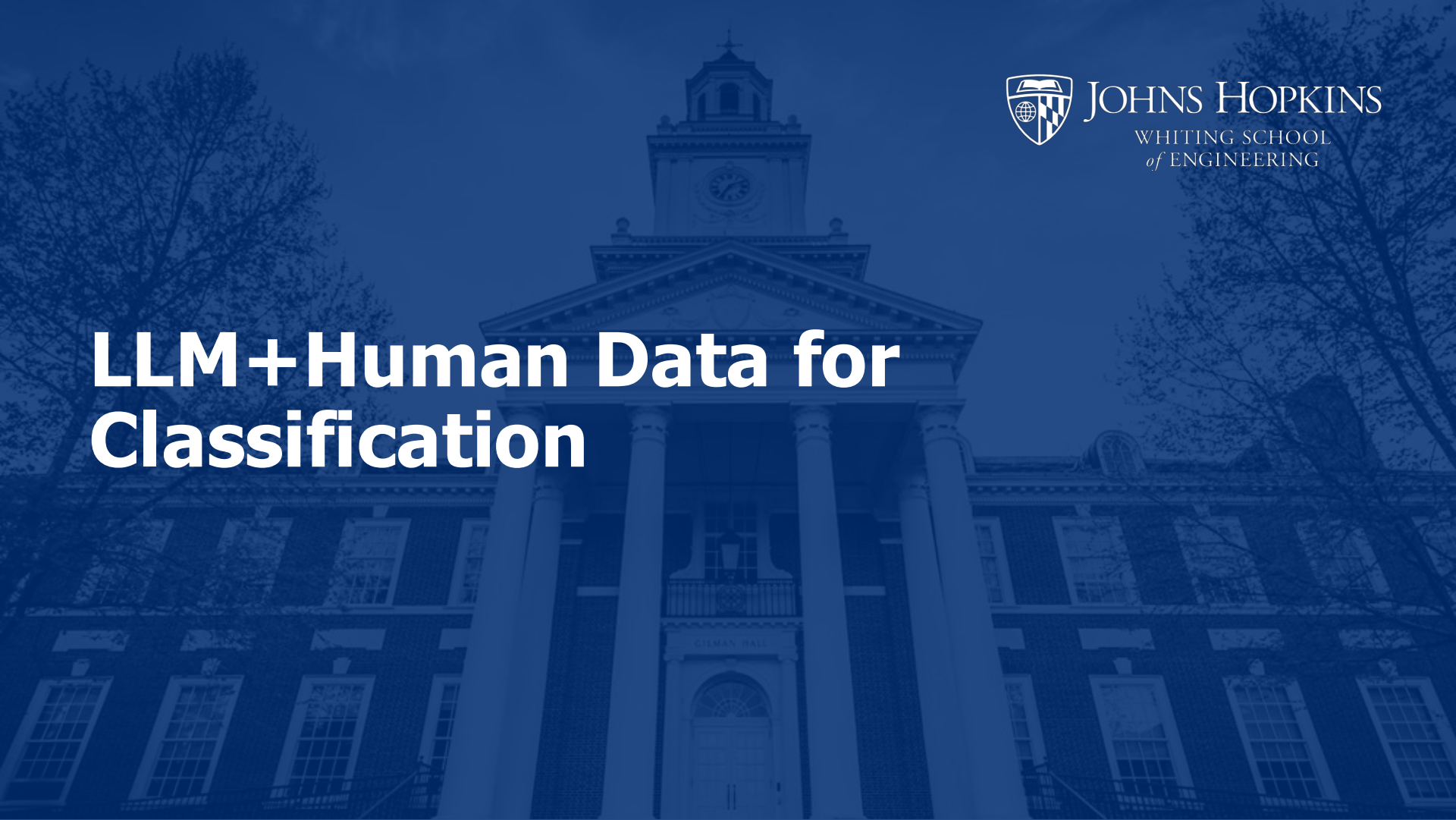


LLM+Human Data for Classification

Recap

- Last class:
  - LLMs/SAEs for corpus analysis
- This class:
  - LLMs as classifiers and data labelers
- Next class:
  - Social simulations: using LLMs to simulate people

Fine-tuning approaches

- If we have 5-10 labeled examples we might use them for few-shot examples
- If we have 100-1000s of examples, what can we do with them?
- Option 1: Fine-tuning the LLM
  - We fine-tuned models like BERT and RoBERTa but newer models are orders of magnitude larger. Can we actually update the model parameters?
- Option 2: Combining LLM and human labels (correcting LLM labels)

Fine-tuning

Parameter-efficient Fine-tuning

- In fine-tuning we need to updating and storing all the parameters of the LM
  - We would need to store a copy of the LM for each task

- With large models, storage management becomes difficult
  - E.g., A model of size 170B parameters requires ~340Gb of storage
  - If you fine-tune a separate model for 100 tasks:
    - 340 * 100 = 34 TB of storage!

Slide thanks: Daniel Khashabi

Parameter-efficient Fine-tuning

- Augmenting the existing pre-trained model with extra parameters or layers and training only the new parameters
  - "parameter efficient": we only update a smaller set of parameters

- Two commonly used methods:
  - Soft prompts
  - Adapters

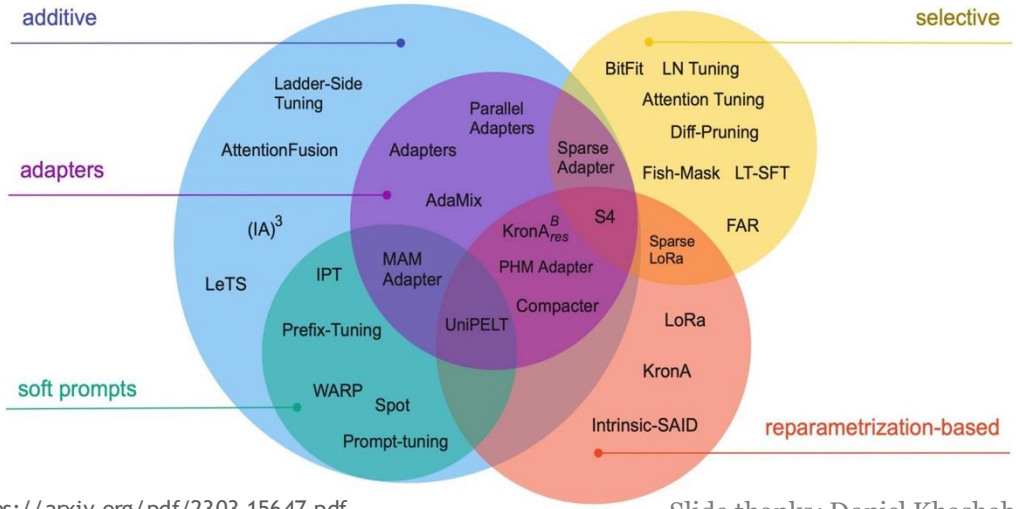
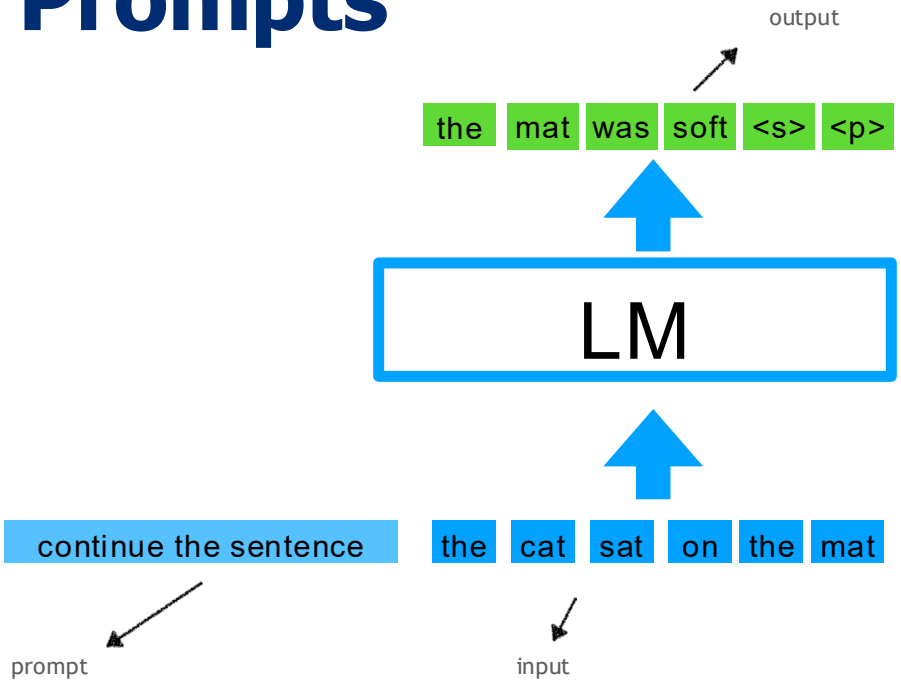


fig source https://arxiv.org/pdf/2303.15647.pdf

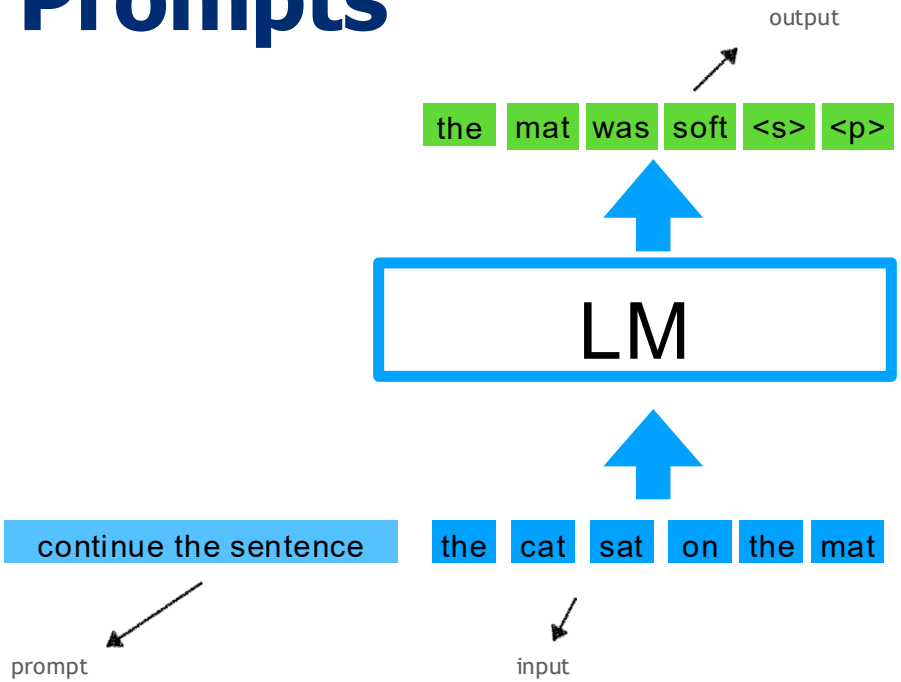
Slide thanks: Daniel Khashabi

Soft Prompts



[Slide: Arman Cohan]

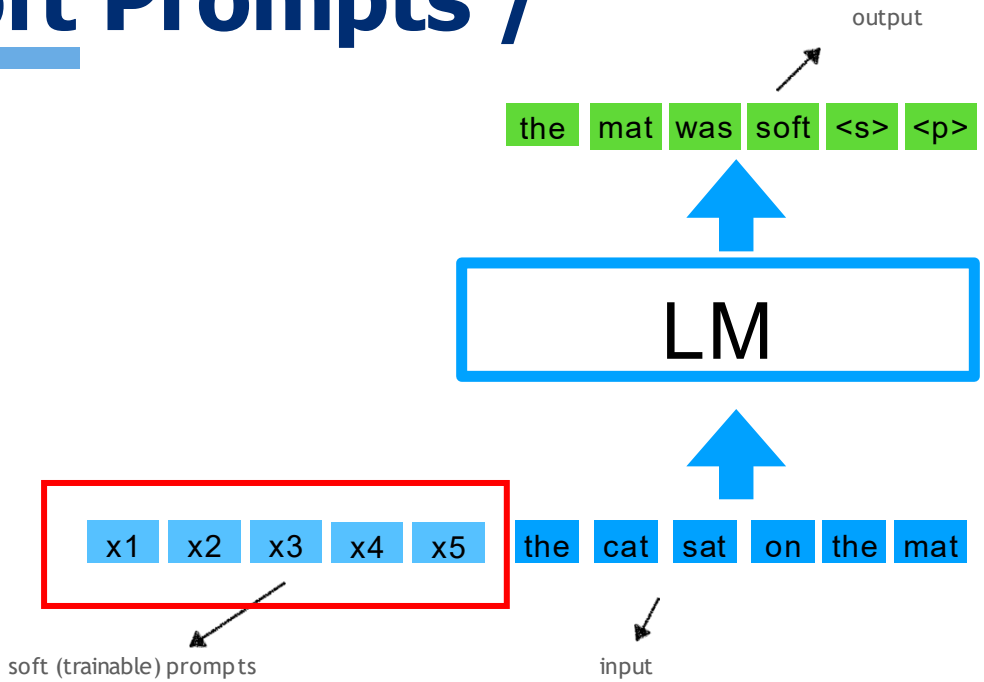
Soft Prompts



Previously, we constructed prompts following "good practice" guidelines and tried paraphrases of them

[Slide: Arman Cohan]

Soft Prompts /

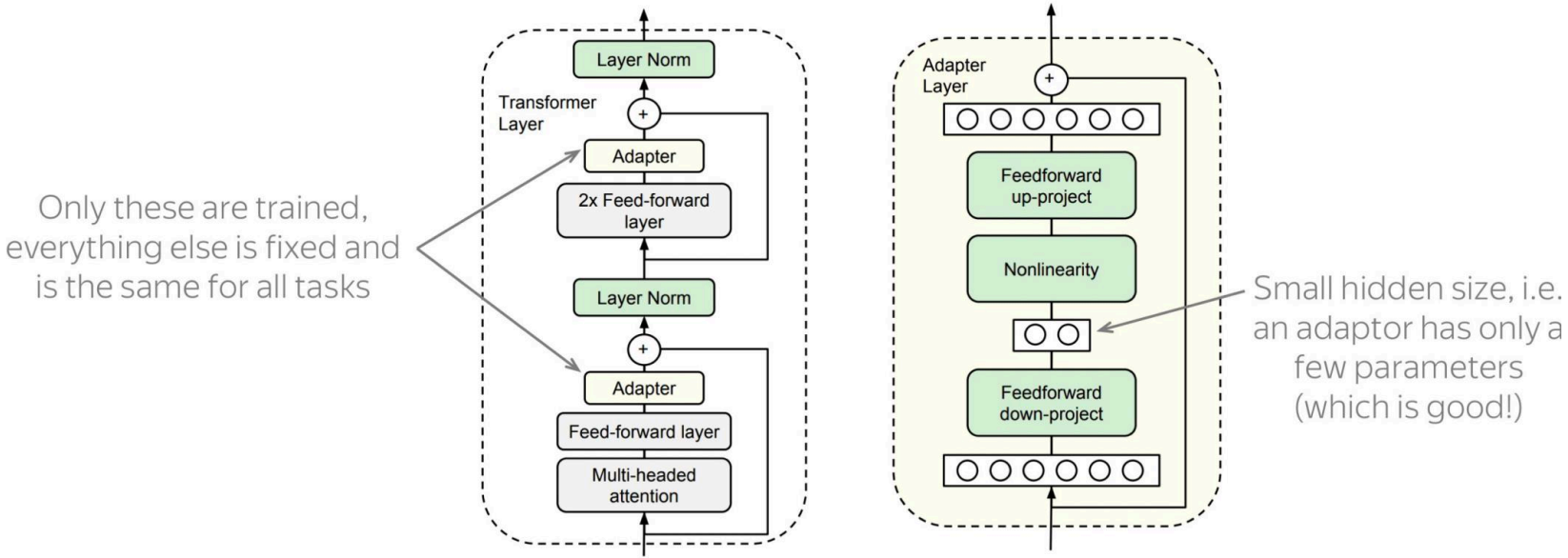


Instead, we can just directly optimize for the best prompt!

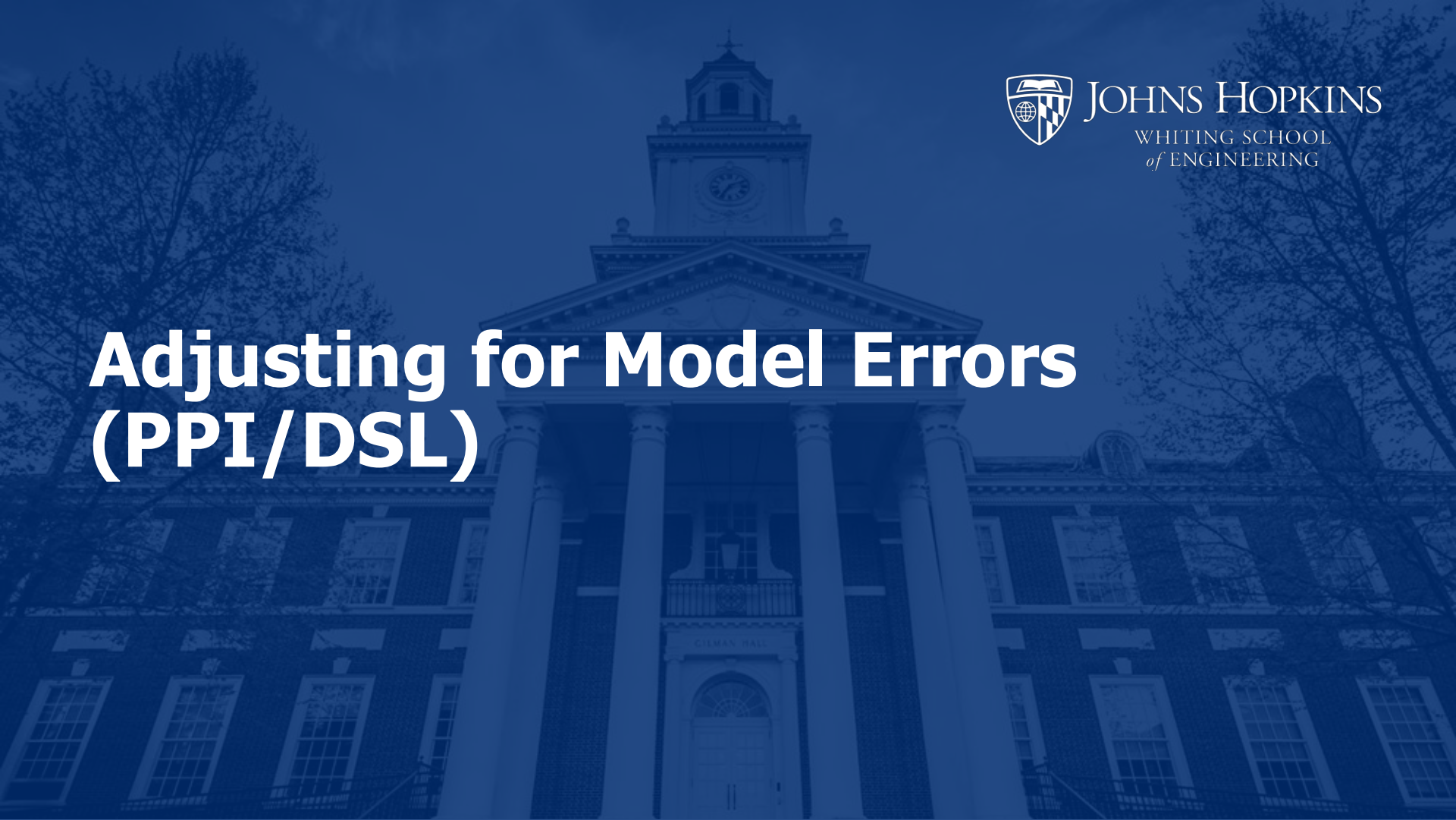
[Slide: Arman Cohan]

Adapters

- **Idea:** train small sub-networks and only tune those.
  - FF projects to a low dimensional space to reduce parameters.
- No need to store a full model for each task, **only the adapter params**.



["Parameter-Efficient Transfer Learning for NLP", Houlsby et al., 2019.]



Adjusting for Model Errors (PPI/DSL)

How good is good enough?

- We saw last week that LLMs can be useful zero or few shot models for some tasks, but performance can be much worse than supervised models
  - Fine-tuning can improve performance

- Do we we actually need to improve performance: If an LLM has accuracy 82% and a supervised model has accuracy 84%, is it worth hours of data annotating for an extra 2%?

- What are we actually trying to do with the generated labels?

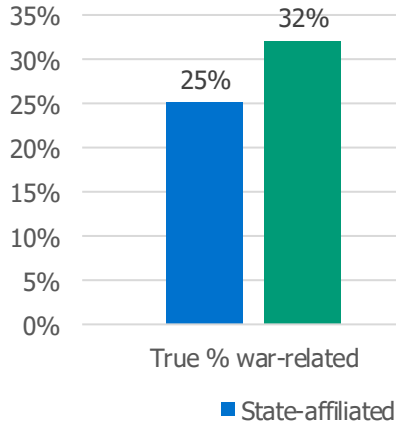
- In CSS, we typically want to use them to learn something about society from our data

How good is good enough?

- Example: Do state-affiliated or independent news outlets in Russia post more often about the Russia-Ukraine war?
  - We use a classifier (e.g. am LLM) to label articles as about the war or not
  - We compute prevalence in state-affiliated vs. independent news outlets
  - If 85% accurate, do we trust the results?
  - What if our classifier is 65% accurate?

Problem: Non-random prediction errors

- Maybe our classifier is 85% accurate overall, but let's pretend most of the classifier mistakes are labeling articles from independent outlets as not war-related (e.g. false negatives):



- We would reach the wrong conclusions because prediction errors are non-random
- We've seen plenty of examples of non-random prediction errors, e.g. hate speech classifiers tend to mislabel posts with identity terms as toxic

Two similar frameworks

- *Prediction Powered Inference (PPI)*
  - Anastasios N. Angelopoulos *et al.* Prediction-powered inference. *Science* **382**,669-674(2023). DOI:10.1126/science.adi6000
  - Proposed by ML researchers
  - Broadly motivates the problem as minimizing the size of confidence intervals
  - Python implementation
- Design-based supervised learning (DSL
  - Egami, Naoki, et al. "Using imperfect surrogates for downstream inference: Design-based supervised learning for social science applications of large language models." *Advances in Neural Information Processing Systems* 36 (2023): 68589-68601.
  - Proposed by political science and sociology researchers
  - Broadly motivates the problem as producing unbiased estimates and valid confidence intervals
  - R implementation

Broad idea

- Labeled data $X = (X_1, X_2, \ldots, X_n)$ and Y= $(Y_1, Y_2, \ldots, Y_n)$
- Much larger set of unlabeled data $\tilde{X} = (\widetilde{X_1}, \widetilde{X_2}, \ldots \widetilde{X_N})$ with predicted labels $\tilde{Y} = (Y, \ldots \widetilde{Y_N})$, where prediction model is independent from X and $\tilde{X}$ (e.g. trained on different data, labels from an off-the-shelf LLM)
- Key idea: Use trusted human annotations to adjust less-trusted predicted annotations in downstream analysis model
- [Could also be expert human vs. crowdsourced rather than human vs. machine]

Example: Mean Estimation

- Example: mean estimation: what proportion of news posts are war-related?
  - Define $Y_i = 0$ if the post is not war-related and $Y_i = 1$ if the post is war-related
  - $\theta^* = E[Y_i]$
- We might normally estimate this by labeling a subset of the data and computing an average of the labeled data:

$$\theta^{classical} = \frac{1}{n}\sum_{i=1}^{n} Y_i$$

  - We could construct a 95% confidence interval as:
  - $\theta^{classical} \pm 1.96\sqrt{\frac{\hat{\sigma}_Y^2}{n}}$ where $\hat{\sigma}_Y^2$ is the estimated variance of Y
  - This interval can be quite large

Example: Mean Estimation

- Example: mean estimation: what proportion of news posts are war-related?
  - Define $Y_i = 0$ if the post is not war-related and $Y_i = 1$ if the post is war-related
  - $\theta^* = E[Y_i]$
- We could try to estimate this value using only the machine-labeled data:

$$\theta^{machine} = \frac{1}{n}\sum_{i=1}^{n} \tilde{Y}_i$$

  - If the classifier has non-random errors, our estimator is biased and the constructed confidence interval is invalid (does not contain the true value 95% of the time)

Example: Mean Estimation

$$\theta^{classical} = \frac{1}{n}\sum_{i=1}^{n} Y_i \qquad \theta^{machine} = \frac{1}{n}\sum_{i=1}^{n} \tilde{Y}_i$$

- **[If our human-labeled data is a random sample]**

- $\theta^{PP} = \underbrace{\frac{1}{N}\sum_{i=1}^{N} \tilde{Y}_i}_{\text{Estimator using the predicted labels}} - \underbrace{\frac{1}{n}\sum_{i=1}^{n}(\tilde{Y}_i - Y_i)}_{\text{“Rectifier” corrects the estimate}}$

Example: Mean Estimation

$$\theta^{PP} = \frac{1}{N}\sum_{i=1}^{N}\tilde{Y}_i \quad - \quad \frac{1}{n}\sum_{i=1}^{n}(\tilde{Y}_i - Y_i)$$

- Theorem proved in PPI/DSL papers:
  - θ^{PP} is accurate: $\theta^{PP} \rightarrow \theta^*$ as the data size grows
  - θ^{PP} is "well-behaved": $\theta^{PP} \approx N(\theta^*, \sigma^2)$
  - 95% confidence interval:

$$\theta^{classical} \pm 1.96\sqrt{\frac{\hat{\sigma}_Y^2}{n}}$$

$$\theta^{PP} \pm 1.96\sqrt{\frac{\hat{\sigma}_{\tilde{Y}-Y}^2}{n} + \frac{\hat{\sigma}_{\tilde{Y}}^2}{N}}$$

If we have lots of unlabeled data, N >> n, the first term dominates

Small if the prediction model is good (can be much smaller than the classical estimator)

General quantities of interest

- What if we want run regressions or compute word statistics or other quantities of interest that are not mean estimation?
- ["confidence-driven inference"]

$$\theta^{CDI} = \hat{\theta}\left(X_i, \tilde{Y}_i\right)_{i=n+1}^{N} - \left(\hat{\theta}\left(X_i, \tilde{Y}_i\right)_{i=1}^{n} - \hat{\theta}(X_i, Y_i)_{i=1}^{n}\right)$$

- X_i can be other relevant values, like confounders in a regression
- θ^{CDI} is also accurate and well-behaved

PPI/DSL Extensions

PPI++

$$\theta^{classical} \pm 1.96\sqrt{\frac{\hat{\sigma}_Y^2}{n}} \qquad \theta^{PP} \pm 1.96\sqrt{\frac{\hat{\sigma}_{\tilde{Y}-Y}^2}{n} + \frac{\hat{\sigma}_{\tilde{Y}}^2}{N}}$$

- θ^{PP} could have a wider confidence interval and higher means squared error than $\theta^{classical}$ if the model errors are large (note that θ^{PP} would still be unbiased and CI would still be valid: a worse model leads to a larger confidence interval)
- **Power tuning**

$$\theta^{\lambda} = \lambda\hat{\theta}\left(X_i, \tilde{Y}_i\right)_{i=n+1}^{N} - \left(\lambda\hat{\theta}\left(X_i, \tilde{Y}_i\right)_{i=1}^{n} - \hat{\theta}(X_i, Y_i)_{i=1}^{n}\right)$$

- $\lambda \in [0,1]$ determines how much we weigh machine vs. human annotations.
- The optimal value of λ is proportional to how $\tilde{Y}$ and Y correlate and can be computed explicitly

Angelopoulos, Anastasios N., John C. Duchi, and Tijana Zrnic. "PPI++: Efficient prediction-powered inference." *arXiv preprint arXiv:2311.01453* (2023).

Non-random sampling

- What if our labeled data is not a random sample of the full dataset?
- As long as we know the sampling strategy (specifically, the probability that each datapoint is sampled for labeling), then we can just incorporate those weights:

$$\theta^{CDI} = \hat{\theta}\left(X_i, \tilde{Y}_i, \overline{W}_i\right)_{i=n+1}^{N} - \left(\hat{\theta}\left(X_i, \tilde{Y}_i, W_i\right)_{i=1}^{n} - \hat{\theta}(X_i, Y_i, W_i)_{i=1}^{n}\right)$$

$$W_i = \frac{\mathbb{I}\{labeled\}}{Prob(labeled)}, \overline{W_i} = \frac{1 - \mathbb{I}\{labeled\}}{1 - Prob(labeled)}$$

(each sample is inversely weighted by the probability of the labeling condition it actually received)

Non-random sampling

LinearRegression

```
class sklearn.linear_model.LinearRegression(*, fit_intercept=True,
copy_X=True, tol=1e-06, n_jobs=None, positive=False) #          [source]
```

```
fit(X, y, sample_weight=None)                                    [source]
```

Fit linear model.

Parameters:

X : *{array-like, sparse matrix} of shape (n_samples, n_features)*

Training data.

y : *array-like of shape (n_samples,) or (n_samples, n_targets)*

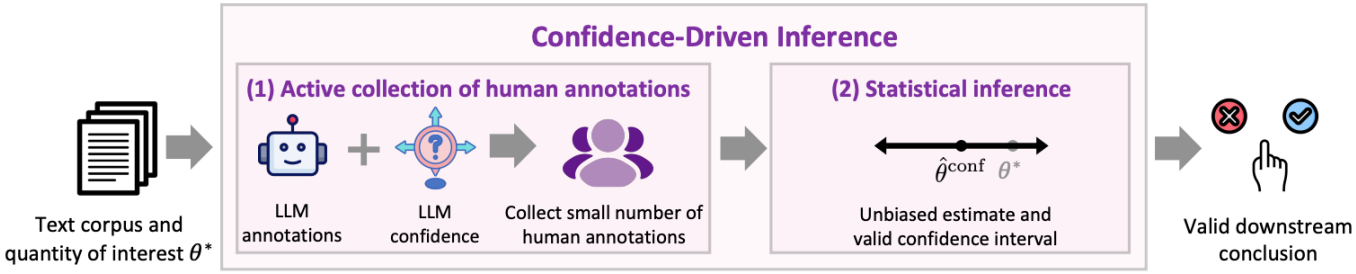
Target values. Will be cast to X's dtype if necessary.

sample_weight : *array-like of shape (n_samples,), default=None*

Individual weights for each sample.

Adaptive Human+LLM Annotations

- Non-random sampling allows us to sample more strategically
- Intuitively, we should have humans label the most difficult examples, ones where the model is worst
- Recall *active learning*: use initial model outputs to guide the next data to annotate
- We can do something similar, but instead of using labeled data to re-train the model, we can use it to improve the statistical inference



Can Unconfident LLM Annotations Be Used for Confident Conclusions? Kristina Gligorić*, Tijana Zrnic*, Cinoo Lee*, Emmanuel Candès, and Dan Jurafsky. NAACL, 2025.

Adaptive Human+LLM Annotations

- It is statistically optimal to have a high probability of labeling datapoints where $\text{err}(\tilde{Y}_i - Y_i)$ largest
  - But we don't know $\text{err}(\tilde{Y}_i - Y_i)$ until we've already generated the human label Y_i
- Proposed solution: LLM verbalized confidence scores are empirically reflective of accuracy

Politeness	**Stage 1**	Is the following text polite? Output either A or B. Output a letter only. A) Polite B) Impolite Text: <text> Answer:
Politeness	**Stage 2**	How likely is it that the following text is <previously provided answer: polite or impolite>? Output the probability only (a number between 0 and 1). Text: <text> Answer:

- We can go one step further and learn a mapping from the LLM-generated confidence score to $\text{err}(\tilde{Y}_i - Y_i)$. Then sample data to annotate based on these predictions of error

Zrnic, Tijana, and Emmanuel J. Candès. "Active statistical inference." *arXiv preprint arXiv:2403.03208* (2024).

Overall procedure

Step 1: Collect LLM predictions $\tilde{Y}_i$ and confidence scores C_i for all documents

Step 2: Collect human annotations for initial document set. Fit mapping from C_i to err($\tilde{Y}_i - Y_i$)

Step 3: Sample documents for human annotations using (reweighted) estimates of err($\tilde{Y}_i - Y_i$)

Step 4: Collect human annotations for sampled documents. Re-estimate the mapping from C_i to err($\tilde{Y}_i - Y_i$)

Step 5: Repeat steps 3-4

Step 6: Compute tuning parameter λ and final estimate:

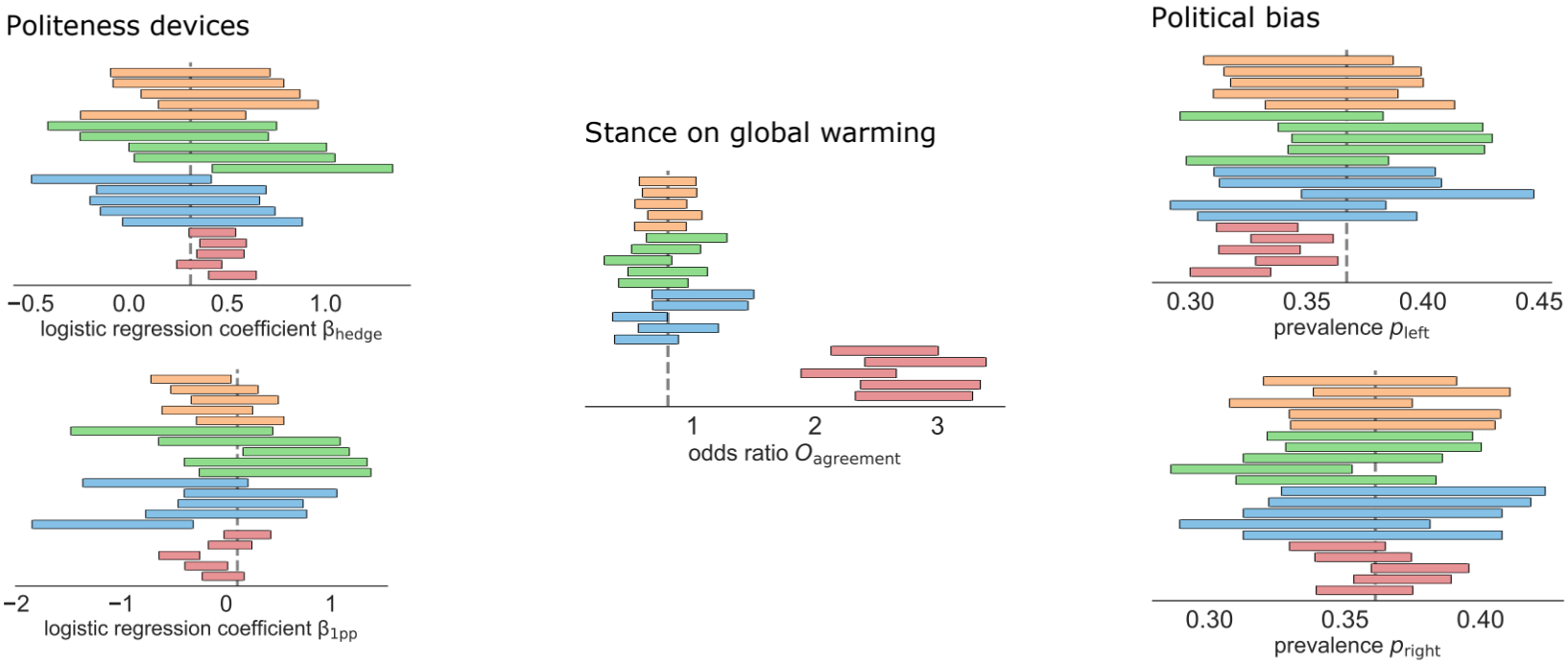
$$\theta^{\lambda} = \lambda\hat{\theta}\left(X_i, \tilde{Y}_i, \overline{W}_i\right)_{i=n+1}^{N} - \left(\lambda\hat{\theta}\left(X_i, \tilde{Y}_i, W_i\right)_{i=1}^{n} - \hat{\theta}(X_i, Y_i, W_i)_{i=1}^{n}\right.$$

Step 7: Compute confidence intervals

Evaluation

- 3 tasks/datasets where the data was entirely human-annotated:
  - We know the true value we are trying to estimate
  - Pretend we didn't label all the data and compare strategies for recovering that estimate
- Politeness:
  - Estimate the impact of linguistic features on perceived politeness (logistic regression coefficient for linguistic features, the dependent variable in the regression is politeness as annotated by humans or LLMs)
- Stance
  - Odds ratio of stance about global warming expressed by news headlines (human/LLM) label vs presence of affirming devices such as "expert" or "proven"
- Political bias
  - Prevalence of left-leaning vs right-leaning articles in a news corpus

Evaluation: coverage

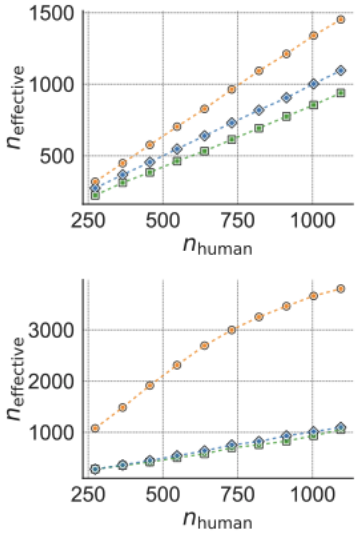


Evaluation: Effective sample size

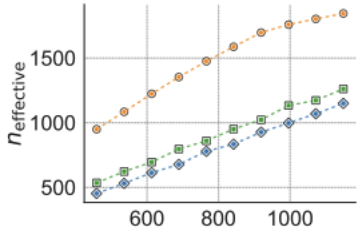
- Alternative evaluation criteria:
  - Given an estimation method (e.g., θ^{λ}), how many data points would we need to annotated to get the same mean squared error under the classical estimator θ^{λ}?
  - ”Effective sample size”: $n_{effective}$
  - $n_{effective} - n_{human}$: how much is using LLM annotations help (or hurting) us under this method?

Evaluation: Effective sample size

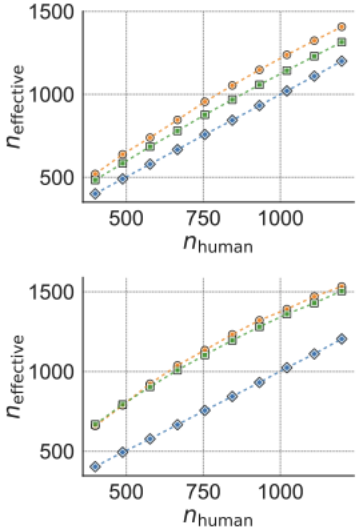
Politeness devices



Stance on global warming



Political bias



confidence-driven human + LLM (non-adaptive) human only LLM only

A different application: image processing (DSL)

- Visual social media (TikTok, Instagram, YouTube, etc) are among the most heavily trafficked social media platforms
- Example (Casas and Webb Williams 2019):
  - Are images a key factor in political mobilization online?
  - Collect images around a Black Lives Matter protest and investigate if emotional content (does this image evoke anger/fear/enthusiasm) effects attention received by the post
  - Manually annotate 9,500 images!
- In LLM/AI era, we could use a vision-language model to generate labels for images similar to LLMs

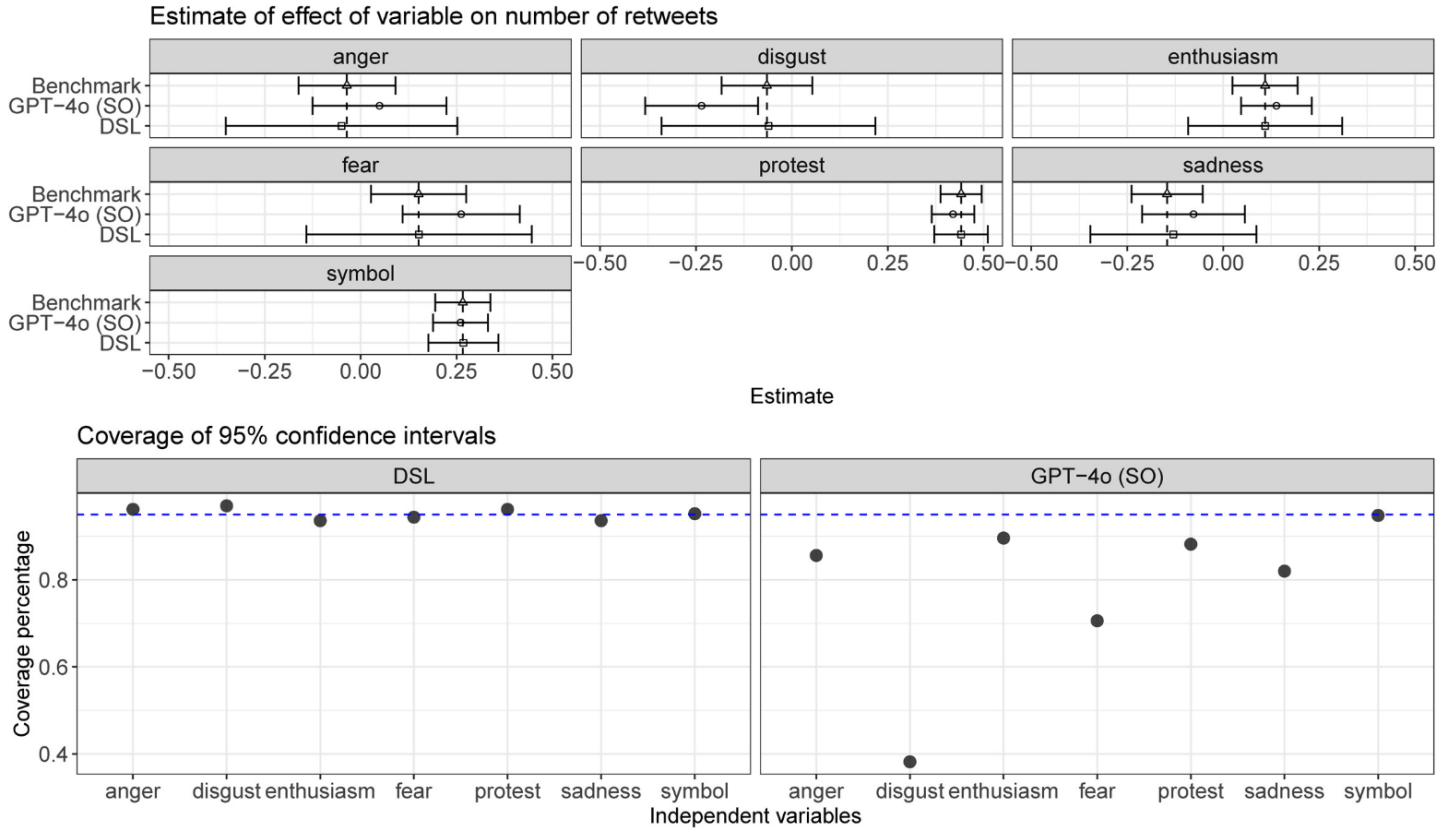
Rister Portinari Maranca, Alessandra, et al. "Correcting the measurement errors of AI-assisted labeling in image analysis using design-based supervised learning." *Sociological Methods & Research* 54.3 (2025): 984-1016.

A different application: image processing

- Use GPT-4o with 100 example images in the context window to automatically annotate the images
- Similar evaluation setup: randomly sample 1200 images to serve as expert annotations and assume remaining images are unannotated

Rister Portinari Maranca, Alessandra, et al. "Correcting the measurement errors of AI-assisted labeling in image analysis using design-based supervised learning." *Sociological Methods & Research* 54.3 (2025): 984-1016.

A different application: image processing



Recap

- Parameter efficient fine-tuning
- Motivation: problem with using LLM annotations (or generally untrusted annotations)
- Formulation of PPI (and DSL)
- Example pipeline combining active annotation, PPI++, and LLM annotations

- What are some open challenges:
  - Formulation for continuous / binary outputs, how would this extend to categorical data?
  - What about models we looked at last week? TopicGPT, LlooM, HiCode? How can we correct errors in those models?
- Logstics

End

Overall procedure

- Annotate data with an LLM
- Using LLM-verbalized confidence scores, select data to label manually
- Compute a *confidence driven* estimate of the value we are actually trying to compute

$$\hat{\theta}^{\text{conf}} = \arg\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} \left(\lambda \hat{\ell}_{\theta,i} + (\ell_{\theta,i} - \lambda \hat{\ell}_{\theta,i}) \frac{\xi_i}{\pi_i} \right)$$

Estimate using LLM annotations

Estimate using human annotations

Indicates if data was human-annotated

Probability data was annotated

[hyperparameter]