Active Learning and Feature-Acquisition with LLMs and Humans

Prabhav ***Singh***[1]*, Haojun* ***Shi***[1]*, Jason* ***Eisner***[1]

Center for Language and Speech Processing
Johns Hopkins University, Baltimore, MD, USA

ELEVATOR PITCH

An **end-to-end transformer-based framework** which will make it **easier and cheaper** to spin up **LLM-based systems [1,2]**, tune, and evaluate them! Imagine you decide to use LLMs-as-a-judge for an annotation task (maybe **de-biased** with human annotations). **You** will have to **decide**:

- Should (and how much of) the annotation be done by a human?
  - If the LLM **is** enough: **Which LLM**?
  - Is CoT needed? Or maybe prefix-tuning?
- Do we need intermediate supervision and **multi-tasking**?
  - Maybe we need intermediate task predictions?
  - Does adding more dimensions help? If so, **which one's?**
  - Absolute ratings or comparative ratings?
- Human annotations are **expensive**!
  - How ***few*** human annotations can we get away with?
  - How can the LLM better mimic the humans: *can the LLM be de-biased?*

Making these decisions heuristically is costly and time-consuming!

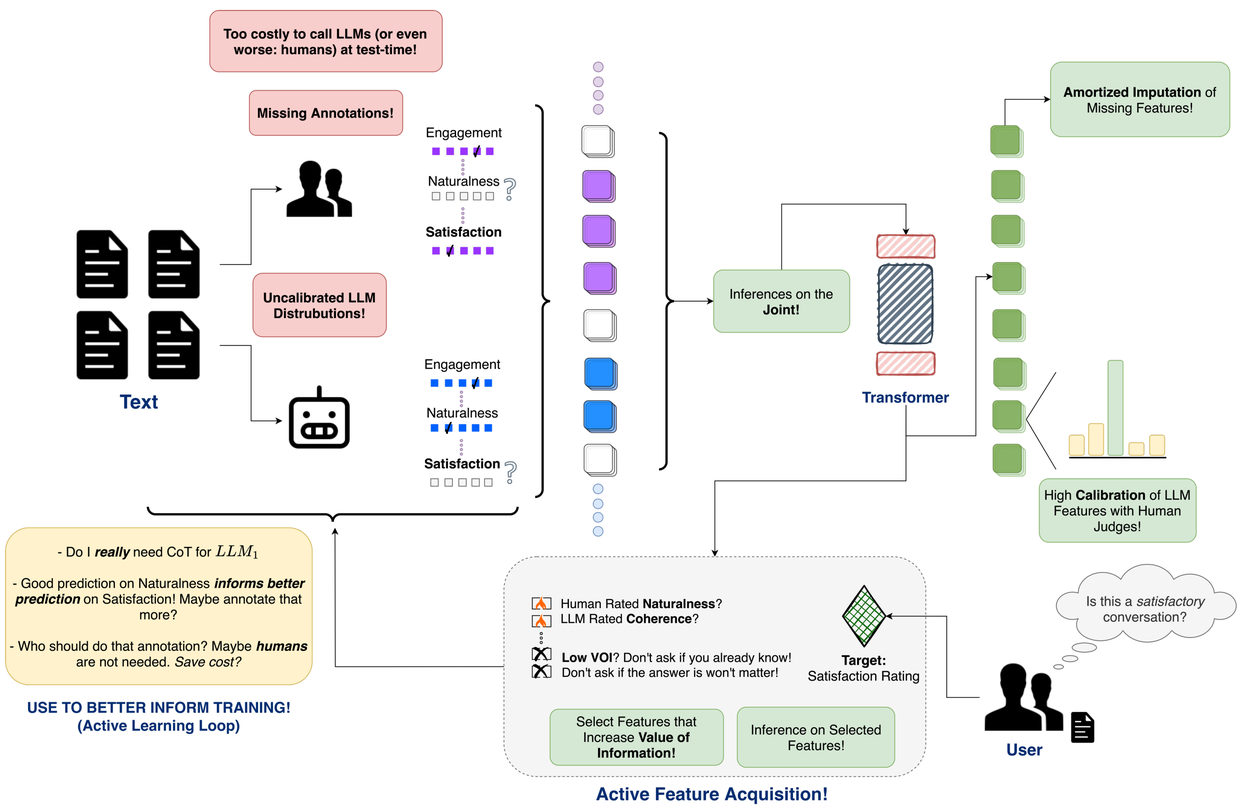
But maybe you don't have to!

Idea 1 (Amortized Imputation): Multidimensional human annotations can be predicted from multidimensional LLM annotations [3]. Extend the same principle to **predict distributions** over **all *uncollected* annotations** (from LLMs and humans) from all annotations that have been collected so far! Sound familiar? ***BERT*** *[4] Style Masked Modeling!*

Idea 2 (Active Feature Acquisition (AFA)): Efficiently characterize each feature. Use **Value of Information [5]** to choose the best feature to *ask* at test-time. Reduce unnecessary expensive calls to LLMs!

Idea 3 (Active Learning): Use AFA to learn parameters that can handle future examples! These can inspire decisions about which features to annotate and **who** should annotate them.

THE SOLUTION

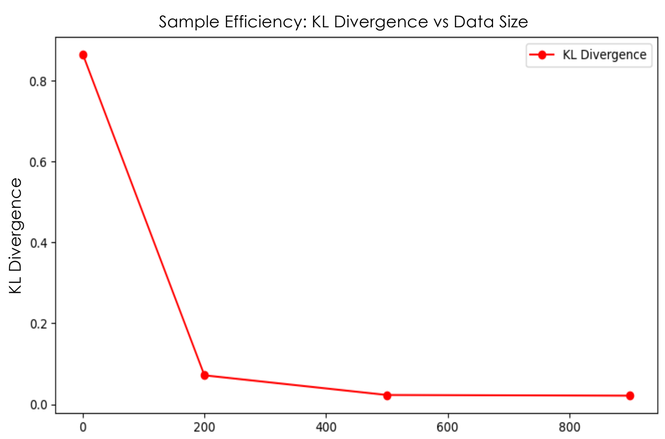


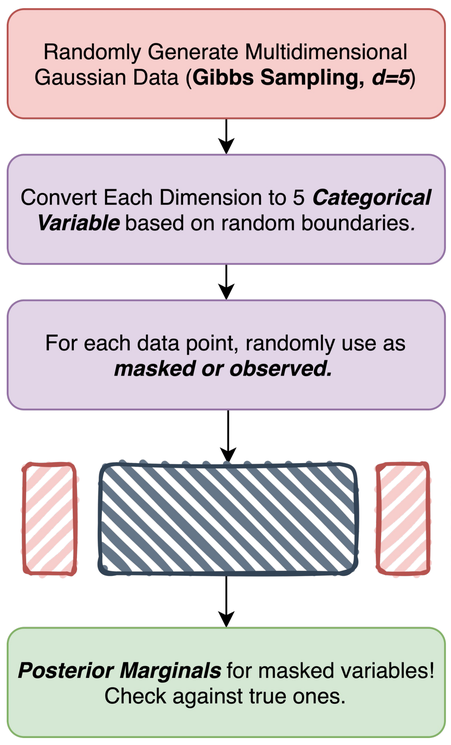
Case Study on Synthetic Data: *Can Transformers Impute Data Efficiently?*

YES!

Table 1. Test Metrics

Metric	Observed Variables	Masked Variables
RMSE	0.0564	0.1737
Pearson Correlation	0.9986	0.9553
Spearman Correlation	0.9435	0.9494



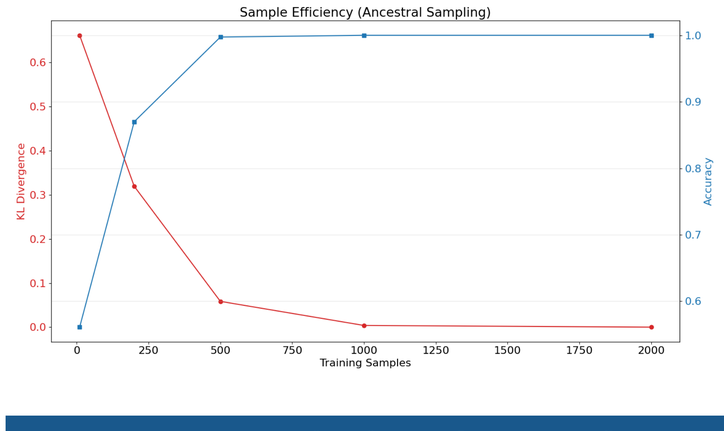


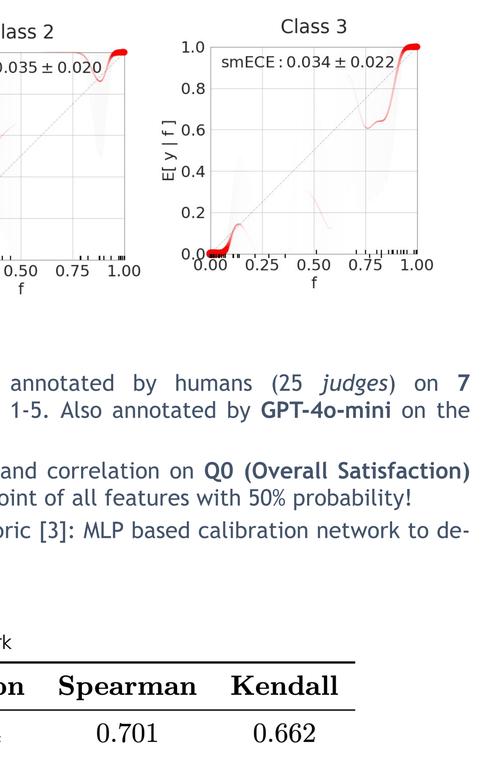
Case Study 2: *Can It Do Graphical Inference*

YES!

Table 2. Test Metrics with Variable Elimination

Metric	Query Variables	Latent Variables
RMSE	0.0219	0.0287
Pearson Correlation	0.9839	0.9921
Spearman Correlation	0.9747	0.9835





De-Biasing The LLM for Better Calibration

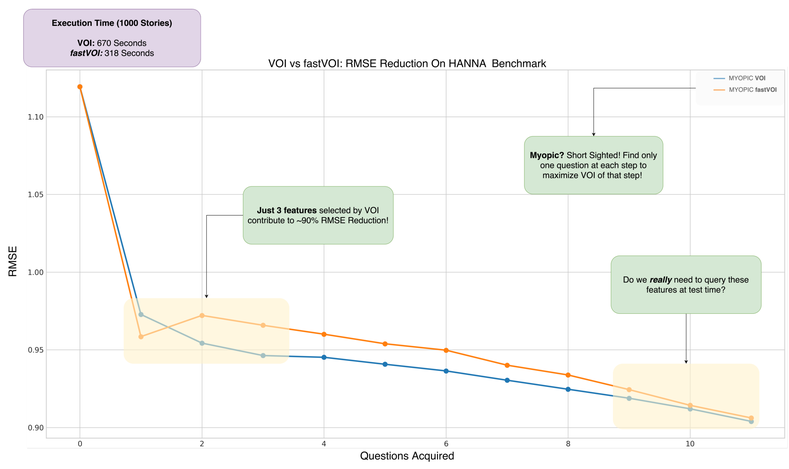
Dataset: **HANNA Benchmark** [5]

1. Subset of 3000 stories annotated by humans (25 *judges*) on **7 dimensions** of a scale of 1-5. Also annotated by **GPT-4o-mini** on the same dimensions.
2. We calculate calibration and correlation on **Q0 (Overall Satisfaction)** while randomly masking joint of all features with 50% probability!
3. Compared against LLMRubric [3]: MLP based calibration network to de-bias the LLM.

Table 3. Comparison of Correlation: HANNA Benchmark

Method	RMSE	Pearson	Spearman	Kendall
LLMRubric (Baseline)	1.695	0.704	0.701	0.662
Ours	**0.280**	**0.970**	**0.973**	**0.953**

(Faster?) Value Of Information: Reduced Cost



ACTIVE FEATURE ACQUISITION (AFA)

"Pick the most helpful feature, reducing uncertainty about a target question efficiently,"

For a target feature Q_0 at index t, a feature at index c, and observed features Q_{obs}, we compute **VOI** as:

$$\mathrm{VOI}(t \mid c, Q_{obs}) = \mathbb{E}_{p(y_c \mid Q_{obs})}\left[\mathcal{L}(y_t \mid Q_{obs}) - \mathcal{L}(y_t \mid Q_{obs}, y_c)\right]$$

Expected reduction in loss L for the target feature if we were to use candidate feature.

Choose candidate with **highest VOI**!

fastVOI: ***The Gradient Trick!***

Compute Loss $L(y_t \mid Q_{obs})$ and **backprop** to get gradient $g_k = \Delta_{p_k} L$.

Estimate loss change for each one-hot p_k^* as $g_k \cdot (p_k^* - p_k)$, average over p_k^* weighted by p_k

Do it for **all k in one backprop pass!**

Under Construction: Planned Work!

Plans to extend our work include

- Explore methods that can allow us to learn a **better policy** for **VOI.** Myopic Policy works! But maybe a reinforcement learning inspired policy works better?
- LLMs give us a distribution over the possible logits for any query. Maybe **making an assumption of a distribution over that distribution** and learning the parameters of the same can improve our '*de-biasing*'? *(Spoiler Alert: Initial experiments have shown that this works for* ***Dirichlet*** *and* ***Logistic Normal*** *for example!)*
- We hope to extend the framework to **Active Learning:**
  - Currently, our AFA framework effectively **selects informative features** within ***each example***.
  - We plan to apply similar principles **across examples**, allowing us to strategically select **which features to observe during training.** This can guide feature observation decisions within training minibatches to improve model parameters, resulting in more accurate predictions on new, unseen minibatches.

One Page with References

[1]{psingh54, hshi33}@jhu.edu, jason@cs.jhu.edu