FANTASTICALLY ORDERED PROMPTS AND WHERE TO FIND THEM: OVERCOMING FEW-SHOT PROMPT ORDER SENSITIVITY

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, Pontus Stenetorp

Presented by: Sarkar Snigdha Sarathi Das

IN-CONTEXT LEARNING (ICL)

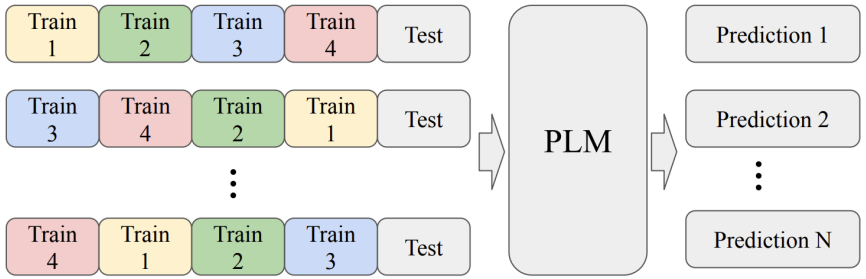
- Two samples:
  - *Review: the greatest musicians. Sentiment: positive*.
  - *Review: redundant concept. Sentiment: negative*
  - Test Sample: *Review: Amazing movie. Sentiment:_____*
- Priming by concatenation
- *Review: the greatest musicians. Sentiment: positive. Review: redundant concept. Sentiment: negative. Review: Amazing movie. Sentiment: _____*

IN-CONTEXT LEARNING (ICL)

- *Review: the greatest musicians. Sentiment: positive. Review: redundant concept. Sentiment: negative. Review: Amazing movie. Sentiment: _____*
- Popularized by GPT-3
- No finetuning/gradient updates required!
- Input length during inference increases
- Highly dependent on the template/prompt

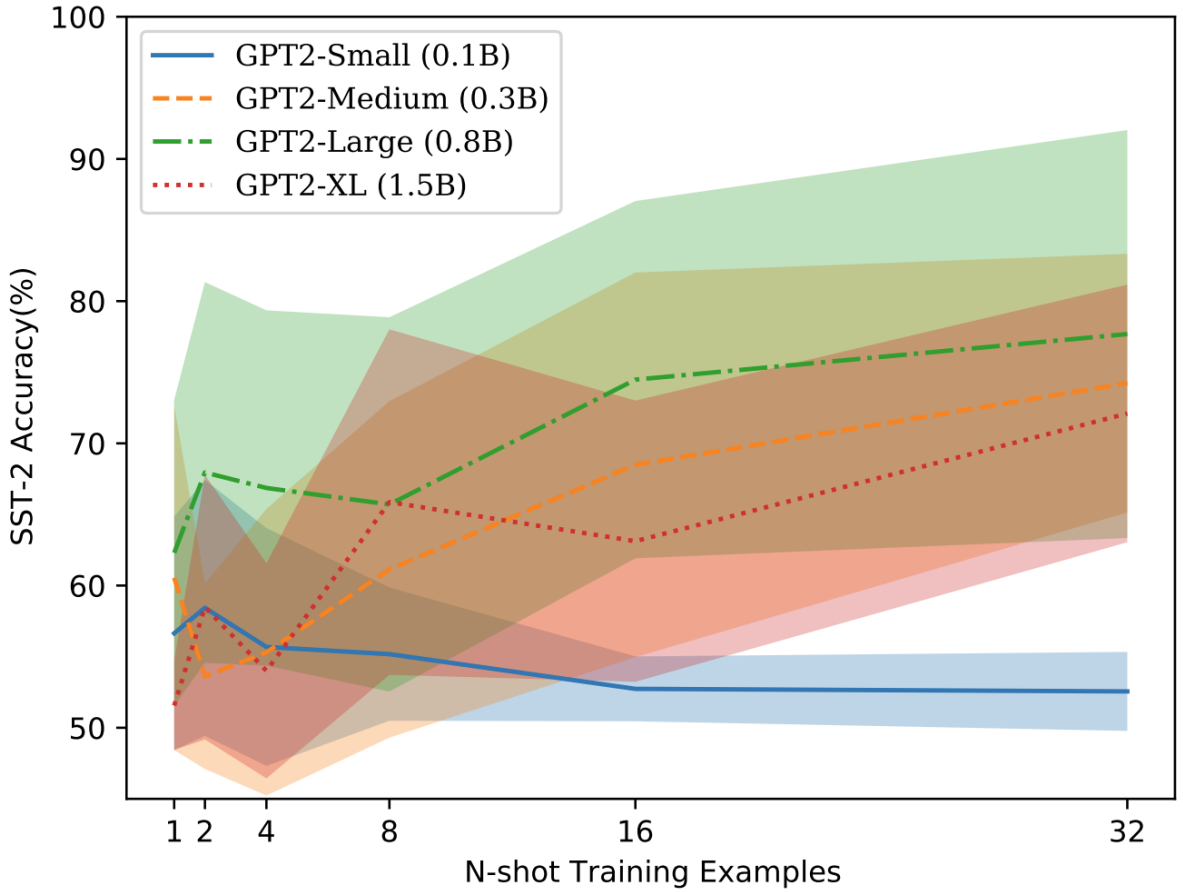
IN-CONTEXT LEARNING (ICL) – ORDER SENSITIVITY

- *Review: the greatest musicians. Sentiment: positive. Review: redundant concept. Sentiment: negative. Review: Amazing movie. Sentiment: _____*
- *Review: redundant concept. Sentiment: negative. Review: the greatest musicians. Sentiment: positive. Review: Amazing movie. Sentiment: _____*
- Unfortunately, this ordering can cause significant variance in performance.



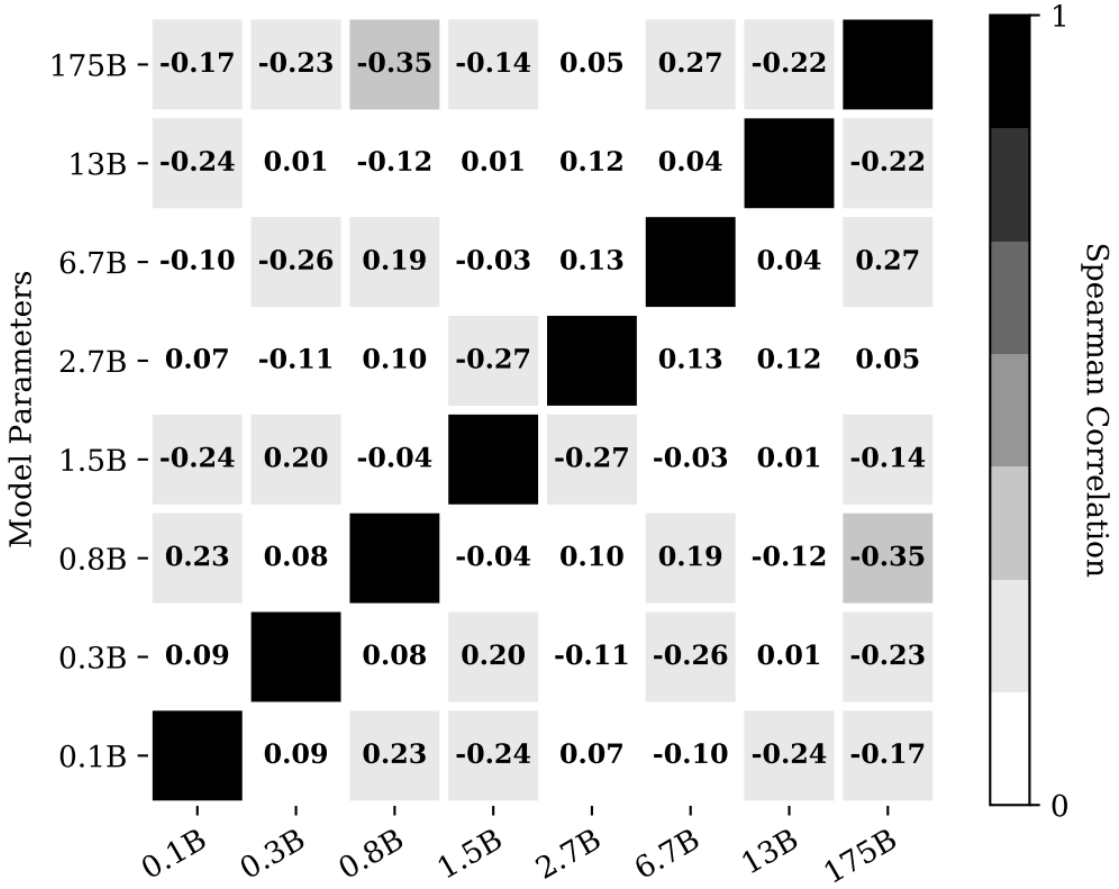
IMPACT OF TRAINING SAMPLE AND MODEL SIZE

- Large models can get great performance, however cannot guarantee low variance in permutations
- Adding more training samples does not reduce variance significantly



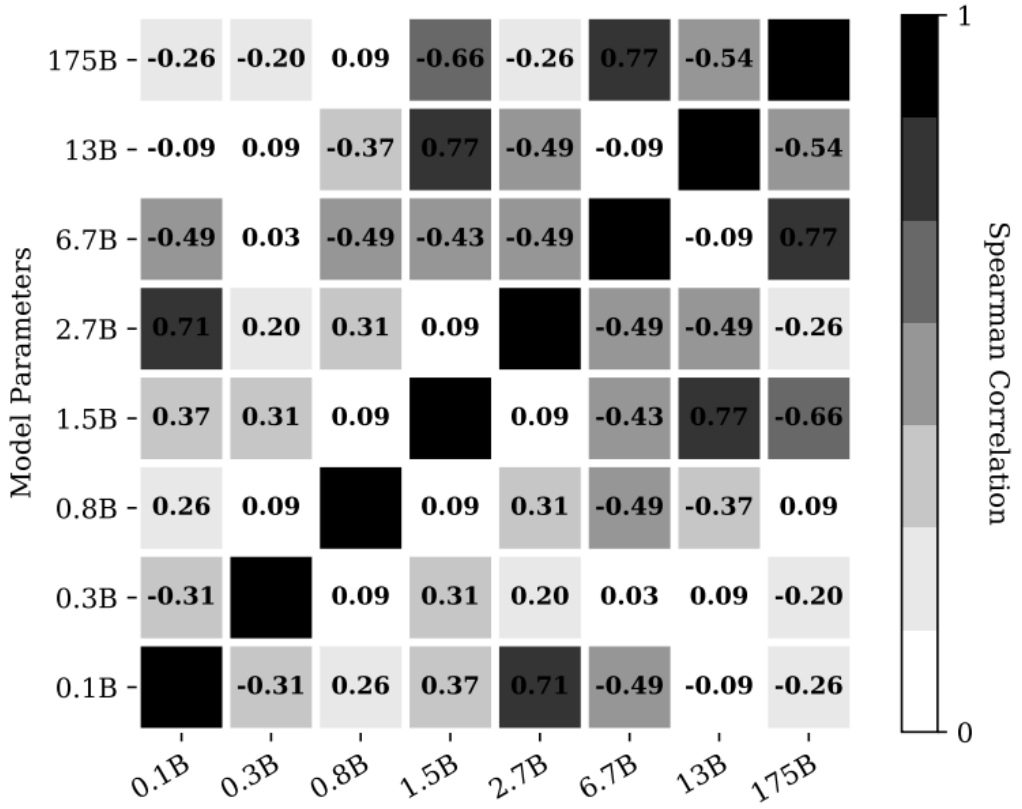
ARE PROMPTS TRANSFERRABLE ACROSS MODELS?

- A specific permutation's performance may drop from 88.7% to 51.6% by changing the underlying model from GPT2-XL (1.5B) to GPT2-Large (0.8B)
- Taken 4 samples – all 24 permutations of prompts and then on prediction -> calculate pairwise Spearman's rank correlation coefficient.
- No correlation at all



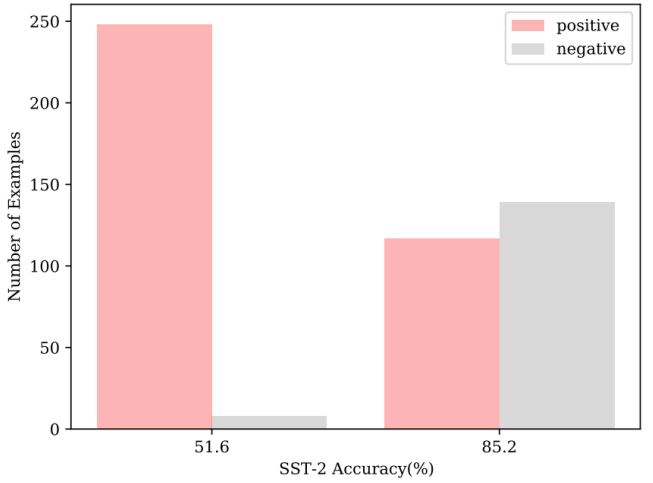
LABEL ORDERINGS ALSO DON'T MATTER ACROSS MODELS

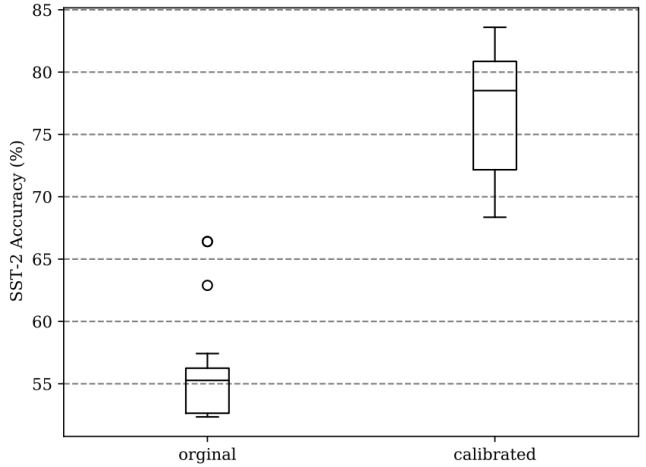
- Measured across six label patterns: NNPP, NPNP, NPPN, PNNP, PNPN, PPNN
- Seems like random behavior



DEGENERATE BEHAVIOUR OF BAD PROMPTS

- Most of the failing prompts suffer from highly unbalanced predicted label distribution
- A way can be using output distribution calibration (Zhao et al. 2021)[1]
- Basically, using N/A as prompt and see output distribution, then add one affine transformation to counter the imbalance
- While that improves performance, variance seems still high across different orders of prompts



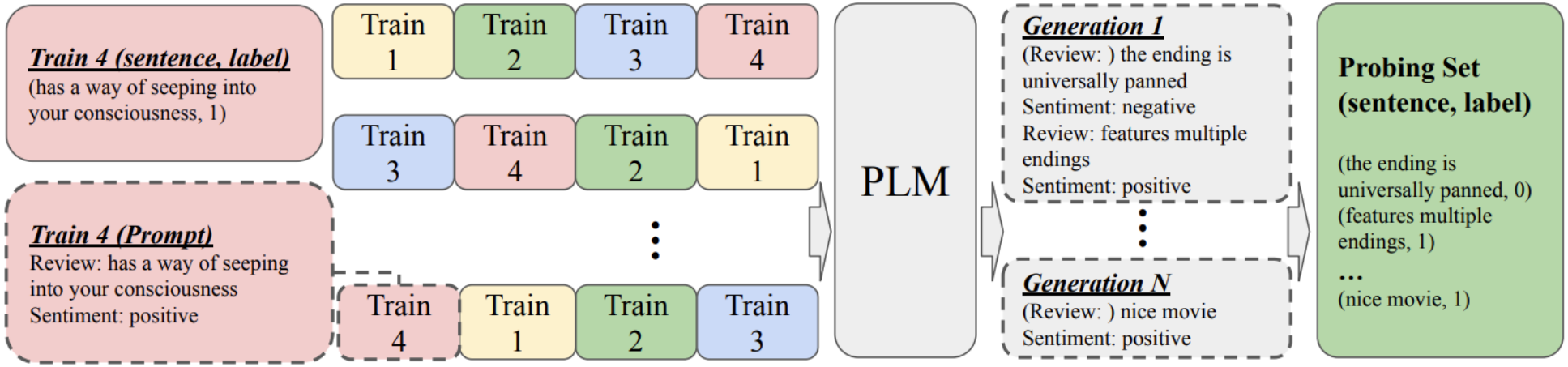


[1] Calibrate Before Use: Improving Few-Shot Performance of Language Models

HOW TO DETERMINE THE BEST ORDERING?

- Simple way: use a dev set.
- Violates True Few shot setting where we don't use any dev set.
- Can we generate a *probing set* by querying the language model instead? Basically, we are using probing set as a substitute for dev set

PROBING SET GENERATION



There is no guarantee of the validity of the labels of these artificial samples!

PROBING METRICS

- Two different methods are used to select best prompt orderings using the generated/probed samples (minus the labels)
  - *Global Entropy*
  - *Local Entropy*

PROBING METRICS: GLOBAL ENTROPY

$$\hat{y}_{i,m} = \underset{v \in V}{\operatorname{argmax}} P(v | c_m \oplus \mathcal{T}(x_i^{'}); \theta)$$

V -> set of labels
c_m -> the context found from current selected ordering
x_i' -> Simulated sample (except the label)

$$p_m^v = \frac{\sum_i \mathbb{1}_{\{\hat{y}_{i,m}=v\}}}{|D|}$$

$$\text{GlobalE}_m = \sum_{v \in V} -p_m^v \log p_m^v$$

PROBING METRICS: GLOBAL ENTROPY

$$\hat{y}_{i,m} = \underset{v \in V}{\operatorname{argmax}} P(v | c_m \oplus \mathcal{T}(x_i^{'}); \theta)$$

V -> set of labels
c_m -> the context found from current selected ordering
x_i' -> Simulated sample (except the label)

$$p_m^v = \frac{\sum_i \mathbb{1}_{\{\hat{y}_{i,m}=v\}}}{|D|}$$

$$\text{GlobalE}_m = \sum_{v \in V} -p_m^v \log p_m^v$$

Thus, we calculate the global Entropy for a context permutation *m*

(allows us to combat extremely unbalanced prediction)

PROBING METRICS: LOCAL ENTROPY

$$p_{i,m}^{v} = P_{(x_i^{'},y_i^{'})\sim D}(v|c_m \oplus \mathcal{T}(x_i^{'});\theta), v \in V$$

$$\text{LocalE}_m = \frac{\sum_i \sum_{v \in V} -p_{i,m}^{v} \log p_{i,m}^{v}}{|D|}$$

Motivation: if a prompt is overly confident, it is likely that it is not behaving as desired/poorly calibrated

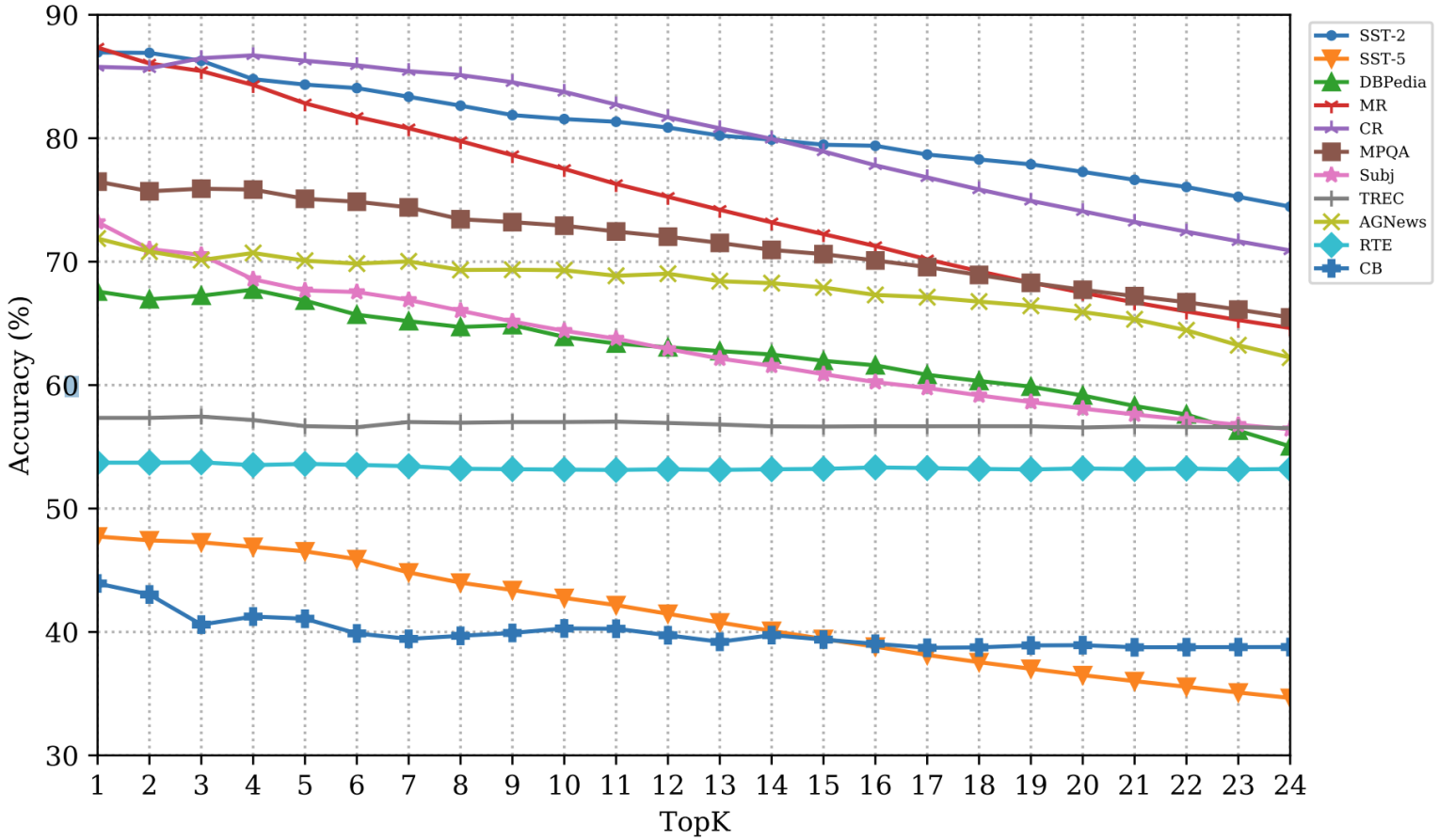
SUMMARY: FROM PROBING TO USING ENTROPY

- Find probe samples for each of the orderings (artificial x’ s)
- Using those x’ s calculate either Global or Local entropy for each of the orderings
- Select top k permutations having highest entropy -> Performant prompts
- Performant prompts are used to evaluate performance in different datasets

MAIN RESULTS

	SST-2	SST-5	DBPedia	MR	CR	MPQA	Subj	TREC	AGNews	RTE	CB
Majority	50.9	23.1	9.4	50.0	50.0	50.0	50.0	18.8	25.0	52.7	51.8
Finetuning (Full)	95.0	58.7	99.3	90.8	89.4	87.8	97.0	97.4	94.7	80.9	90.5
GPT-2 0.1B	$58.9_{7.8}$	$29.0_{4.9}$	$44.9_{9.7}$	$58.6_{7.6}$	$58.4_{6.4}$	$68.9_{7.1}$	$52.1_{0.7}$	$\mathbf{49.2}_{4.7}$	$50.8_{11.9}$	$49.7_{2.7}$	$50.1_{1.0}$
LocalE	$\mathbf{65.2}_{3.9}$	$34.4_{3.4}$	$53.3_{4.9}$	$66.0_{6.3}$	$\mathbf{65.0}_{3.4}$	$72.5_{6.0}$	$52.9_{1.3}$	$48.0_{3.9}$	$61.0_{5.9}$	$\mathbf{53.0}_{3.3}$	$49.9_{1.6}$
GlobalE	$63.8_{5.8}$	$\mathbf{35.8}_{2.0}$	$\mathbf{56.1}_{4.3}$	$\mathbf{66.4}_{5.8}$	$64.8_{2.7}$	$\mathbf{73.5}_{4.5}$	$\mathbf{53.0}_{1.3}$	$46.1_{3.7}$	$\mathbf{62.1}_{5.7}$	$\mathbf{53.0}_{3.0}$	$\mathbf{50.3}_{1.6}$
Oracle	$73.5_{1.7}$	$38.2_{4.0}$	$60.5_{4.2}$	$74.3_{4.9}$	$70.8_{4.4}$	$81.3_{2.5}$	$55.2_{1.7}$	$58.1_{4.3}$	$70.3_{2.8}$	$56.8_{2.0}$	$52.1_{1.3}$
GPT-2 0.3B	$61.0_{13.2}$	$25.9_{5.9}$	$51.7_{7.0}$	$54.2_{7.8}$	$56.7_{9.4}$	$54.5_{8.8}$	$54.4_{7.9}$	$52.6_{4.9}$	$47.7_{10.6}$	$48.8_{2.6}$	$50.2_{5.3}$
LocalE	$75.3_{4.6}$	$31.0_{3.4}$	$47.1_{3.7}$	$65.2_{6.6}$	$\mathbf{70.9}_{6.3}$	$67.6_{7.2}$	$\mathbf{66.7}_{9.3}$	$53.0_{3.9}$	$51.2_{7.3}$	$\mathbf{51.8}_{1.0}$	$47.1_{4.2}$
GlobalE	$\mathbf{78.7}_{5.2}$	$\mathbf{31.7}_{5.2}$	$\mathbf{58.3}_{5.4}$	$\mathbf{67.0}_{5.9}$	$70.7_{6.7}$	$\mathbf{68.3}_{6.9}$	$65.8_{10.1}$	$\mathbf{53.3}_{4.6}$	$\mathbf{59.6}_{7.2}$	$51.1_{1.9}$	$\mathbf{50.3}_{3.7}$
Oracle	$85.5_{4.3}$	$40.5_{6.3}$	$65.2_{7.6}$	$74.7_{6.1}$	$80.4_{5.4}$	$77.3_{2.3}$	$79.4_{2.4}$	$63.3_{2.9}$	$68.4_{8.0}$	$53.9_{1.3}$	$62.5_{7.4}$
GPT-2 0.8B	$74.5_{10.3}$	$34.7_{8.2}$	$55.0_{12.5}$	$64.6_{13.1}$	$70.9_{12.7}$	$65.5_{8.7}$	$56.4_{9.1}$	$56.5_{2.7}$	$62.2_{11.6}$	$53.2_{2.0}$	$38.8_{8.5}$
LocalE	$81.1_{5.5}$	$40.3_{4.7}$	$56.7_{7.5}$	$82.6_{4.2}$	$85.4_{3.8}$	$73.6_{4.8}$	$\mathbf{70.4}_{4.2}$	$56.2_{1.7}$	$62.7_{8.1}$	$53.3_{1.6}$	$38.4_{5.2}$
GlobalE	$\mathbf{84.8}_{4.1}$	$\mathbf{46.9}_{1.1}$	$\mathbf{67.7}_{3.6}$	$\mathbf{84.3}_{2.9}$	$\mathbf{86.7}_{2.5}$	$\mathbf{75.8}_{3.1}$	$68.6_{6.5}$	$\mathbf{57.2}_{2.3}$	$\mathbf{70.7}_{3.6}$	$\mathbf{53.5}_{1.5}$	$\mathbf{41.2}_{4.5}$
Oracle	$88.9_{1.8}$	$48.4_{0.7}$	$72.3_{3.3}$	$87.5_{1.1}$	$89.9_{0.9}$	$80.3_{4.9}$	$76.6_{4.1}$	$62.1_{1.5}$	$78.1_{1.3}$	$57.3_{1.0}$	$53.2_{5.3}$
GPT-2 1.5B	$66.8_{10.8}$	$41.7_{6.7}$	$82.6_{2.5}$	$59.1_{11.9}$	$56.9_{9.0}$	$73.9_{8.6}$	$59.7_{10.4}$	$53.1_{3.3}$	$77.6_{7.3}$	$55.0_{1.4}$	$53.8_{4.7}$
LocalE	$76.7_{8.2}$	$\mathbf{45.1}_{3.1}$	$83.8_{1.7}$	$78.1_{5.6}$	$71.8_{8.0}$	$78.5_{3.6}$	$69.7_{5.8}$	$53.6_{3.1}$	$79.3_{3.7}$	$\mathbf{56.8}_{1.1}$	$52.6_{3.9}$
GlobalE	$\mathbf{81.8}_{3.9}$	$43.5_{4.5}$	$\mathbf{83.9}_{1.8}$	$\mathbf{77.9}_{5.7}$	$\mathbf{73.4}_{6.0}$	$\mathbf{81.4}_{2.1}$	$\mathbf{70.9}_{6.0}$	$\mathbf{55.5}_{3.0}$	$\mathbf{83.9}_{1.2}$	$56.3_{1.2}$	$\mathbf{55.1}_{4.6}$
Oracle	$86.1_{1.5}$	$50.9_{1.0}$	$87.3_{1.5}$	$84.0_{2.7}$	$80.3_{3.3}$	$85.1_{1.4}$	$79.9_{5.7}$	$59.0_{2.3}$	$86.1_{0.7}$	$58.2_{0.6}$	$63.9_{4.3}$
GPT-3 2.7B	$78.0_{10.7}$	$35.3_{6.9}$	$81.1_{1.8}$	$68.0_{12.9}$	$76.8_{11.7}$	$66.5_{10.3}$	$49.1_{2.9}$	$55.3_{4.4}$	$72.9_{4.8}$	$48.6_{1.9}$	$50.4_{0.7}$
LocalE	$\mathbf{81.0}_{6.0}$	$42.3_{4.7}$	$80.3_{1.7}$	$75.6_{4.1}$	$79.0_{5.5}$	$72.5_{5.8}$	$54.2_{4.2}$	$54.0_{2.6}$	$72.3_{4.6}$	$50.4_{1.9}$	$50.5_{0.8}$
GlobalE	$80.2_{4.2}$	$\mathbf{43.2}_{4.3}$	$\mathbf{81.2}_{0.9}$	$\mathbf{76.1}_{3.8}$	$\mathbf{80.3}_{3.4}$	$\mathbf{73.0}_{4.3}$	$\mathbf{54.3}_{4.0}$	$\mathbf{56.7}_{2.0}$	$\mathbf{78.1}_{1.9}$	$\mathbf{51.3}_{1.8}$	$\mathbf{51.2}_{0.8}$
Oracle	$89.8_{0.7}$	$48.0_{1.1}$	$85.4_{1.6}$	$87.4_{0.9}$	$90.1_{0.7}$	$80.9_{1.4}$	$60.3_{10.3}$	$62.8_{4.2}$	$81.3_{2.9}$	$53.4_{3.1}$	$52.5_{1.4}$
GPT-3 175B	$\mathbf{93.9}_{0.6}$	$54.4_{2.5}$	$95.4_{0.9}$	$\mathbf{94.6}_{0.7}$	$91.0_{1.0}$	$83.2_{1.5}$	$71.2_{7.3}$	$72.1_{2.7}$	$85.1_{1.7}$	$70.8_{2.8}$	$75.1_{5.1}$
LocalE	$93.8_{0.5}$	$\mathbf{56.0}_{1.7}$	$95.5_{0.9}$	$94.5_{0.7}$	$91.3_{0.5}$	$\mathbf{83.3}_{1.7}$	$75.0_{4.6}$	$71.8_{3.2}$	$\mathbf{85.9}_{0.7}$	$\mathbf{71.9}_{1.4}$	$74.6_{4.2}$
GlobalE	$\mathbf{93.9}_{0.6}$	$53.2_{2.1}$	$\mathbf{95.7}_{0.7}$	$\mathbf{94.6}_{0.2}$	$\mathbf{91.7}_{0.4}$	$82.0_{0.8}$	$\mathbf{76.3}_{3.5}$	$\mathbf{73.6}_{2.5}$	$85.7_{1.0}$	$71.8_{1.9}$	$\mathbf{79.9}_{3.3}$
Oracle	$94.7_{0.2}$	58.2	$96.7_{0.2}$	$95.5_{0.2}$	$92.6_{0.4}$	$85.5_{0.8}$	$81.1_{4.9}$	$77.0_{1.2}$	$87.7_{0.6}$	$74.7_{0.4}$	$83.0_{0.9}$

MAIN RESULTS - ROBUSTNESS



COMPARISON AGAINST SUBSET OF TRAINING DATA FOR TUNING

	GPT-2 0.1B	GPT-2 0.3B	GPT-2 0.8B	GPT-2 1.5B
Baseline	$58.9_{7.8}$	$61.0_{13.2}$	$74.5_{10.3}$	$66.8_{10.8}$
LocalE	$\mathbf{65.2}_{3.9}$	$75.3_{4.6}$	$81.1_{5.5}$	$76.7_{8.2}$
GlobalE	$63.8_{5.8}$	$\mathbf{78.7}_{5.2}$	$\mathbf{84.8}_{4.1}$	$\mathbf{81.8}_{3.9}$
Split Training Set	$62.8_{5.3}$	$64.2_{6.1}$	$75.1_{6.8}$	$71.4_{7.8}$

- 4-shot data is split in half : dev and test set
- LocalE and GlobalE still outperforms this technique

COMPARISON AGAINST SUBSET OF TRAINING DATA FOR TUNING

	GPT-2 0.1B	GPT-2 0.3B	GPT-2 0.8B	GPT-2 1.5B
Baseline	$58.9_{7.8}$	$61.0_{13.2}$	$74.5_{10.3}$	$66.8_{10.8}$
LocalE	$\mathbf{65.2}_{3.9}$	$75.3_{4.6}$	$81.1_{5.5}$	$76.7_{8.2}$
GlobalE	$63.8_{5.8}$	$\mathbf{78.7}_{5.2}$	$\mathbf{84.8}_{4.1}$	$\mathbf{81.8}_{3.9}$
Split Training Set	$62.8_{5.3}$	$64.2_{6.1}$	$75.1_{6.8}$	$71.4_{7.8}$

- 4-shot data is split in half : dev and test set
- LocalE and GlobalE still outperforms this technique

CONCLUSION

- This paper effectively shows how few-shot prompts suffer form order sensitivity
- A thorough analysis has shown that order sensitivity is present across tasks, model sizes, prompt templates, samples, number of training samples
- Without using any external dev set, the local and global entropy calculation using probed samples can effectively detect any prompt that may cause imbalance
- Through thorough investigation, on average 13% improvement can be gotten across 11 text classification tasks

CRITIQUE

- The limited context window already severely limits how many samples can be used as context. (Liu et al. 2021[2]). As a result, in practice often few shot examples are filtered out, those can be used as dev set.
- While no finetuning is required, ICL itself is computationally heavy during inference due to huge context window needed (less sparsity)
- Performance still not exactly same as full finetuning even with GPT-3, sometimes quite a bit far away.
- Performance with few shot finetuning could be compared, at least in smaller models which could serve as a good baseline

[2]What makes good in-context examples for GPT-3?

FUTURE WORK

- Apart from ICL, sometimes in finetuning we have trouble finding dev sets in few shot cases since we can use all the few shot samples there. This approach can be adapted and modified to be used in calibrating those models as well, for model selection
- Is it possible to generate **ONE** artificial sample that gives the same effect as all the in-context few-shot examples we have? This can both reduce computational load in in-context learning and effectively use all the few-shot samples to get better performance.