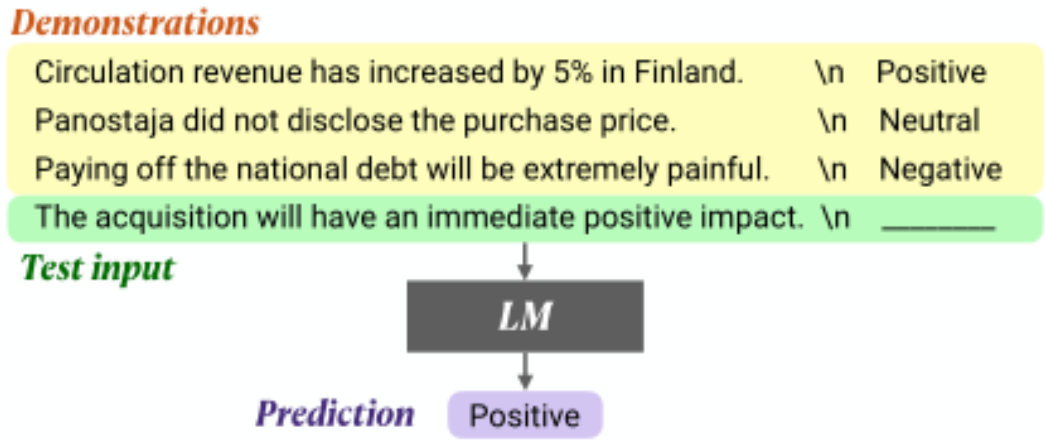
In-context learning and limits I

AntNLP 2022 Fall Seminar

Li Yuqian

In-context learning

- Given $\theta^* \in \Theta$, the prompt is a concatenation of n independent demonstrations and 1 test input x_{test} that are all conditioned on θ^*
- The goal is to predict the test output y_{test} by predicting the next token.



Fantastically Ordered Prompts and Where to Find Them: Overcoming Few-Shot Prompt Order Sensitivity

Yao Lu[†] **Max Bartolo**[†] **Alastair Moore**[‡] **Sebastian Riedel**[†] **Pontus Stenetorp**[†]

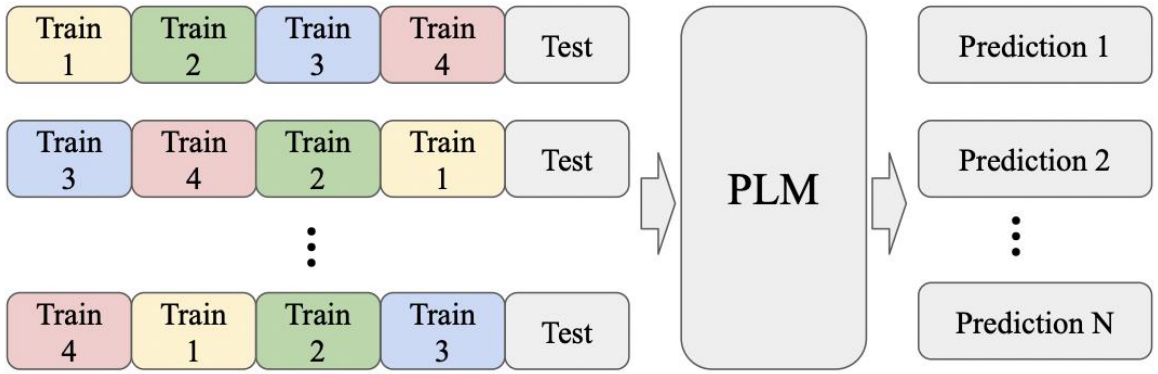
[†]University College London [‡]Mishcon de Reya LLP

{yao.lu,m.bartolo,s.riedel,p.stenetorp}@cs.ucl.ac.uk

alastair.moore@mishcon.com

Prompt Order Sensitivity

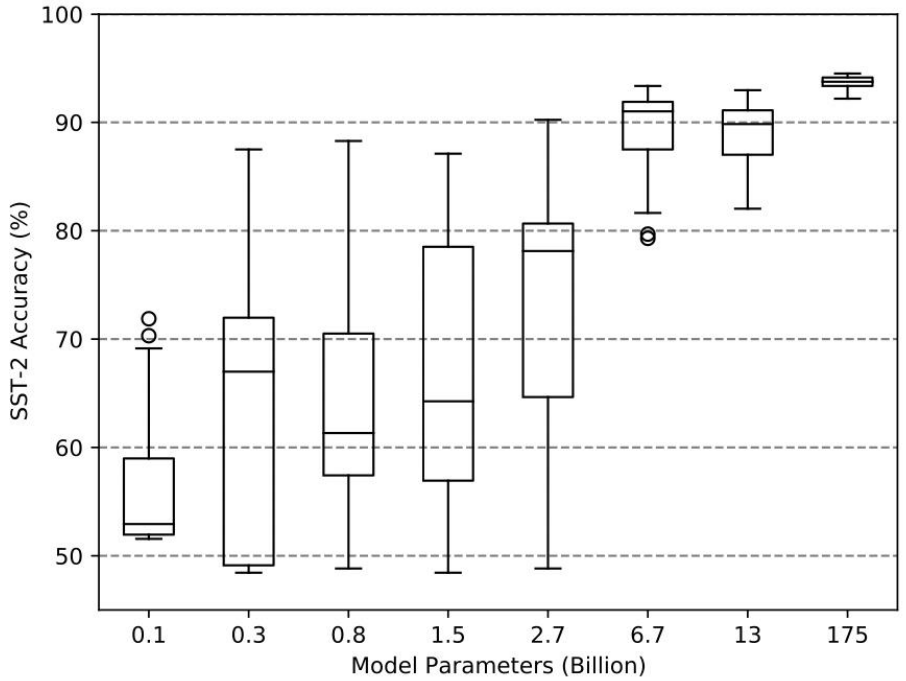
1. Take 4 samples, create all 24 permutations, test prediction performance.

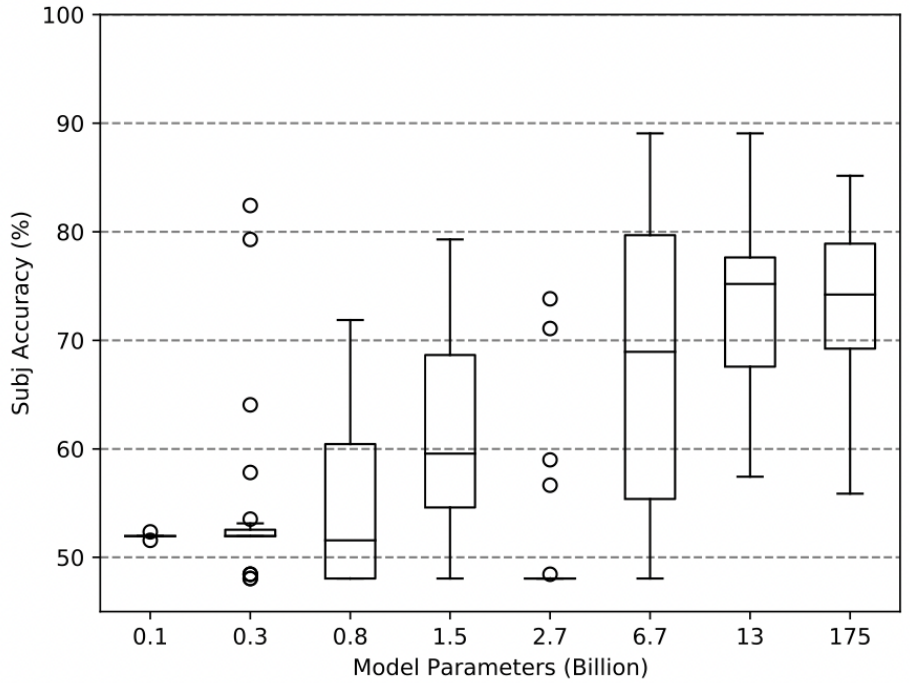


2. Test on various datasts and models (4 GPT-2, 4 GPT-3 sizes)

✍: Aayush Mishra and Lingfeng Shen

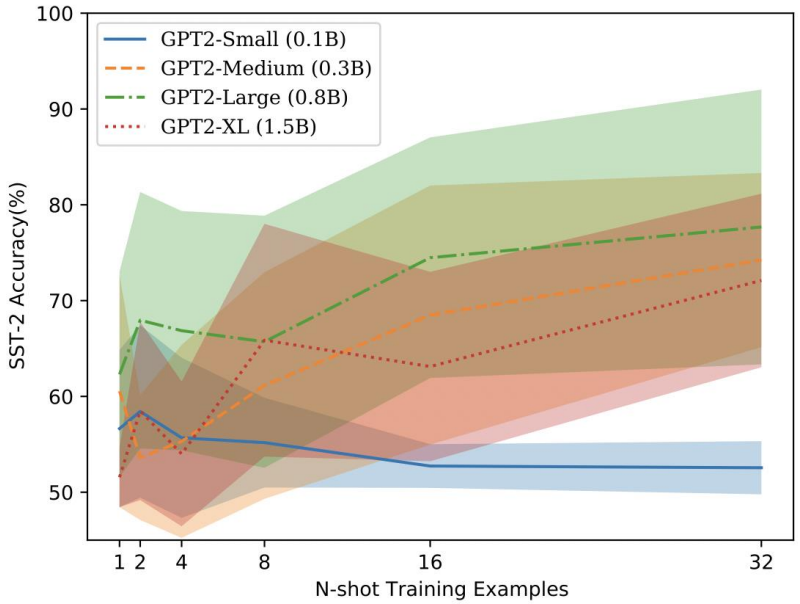
Prompt Design Study



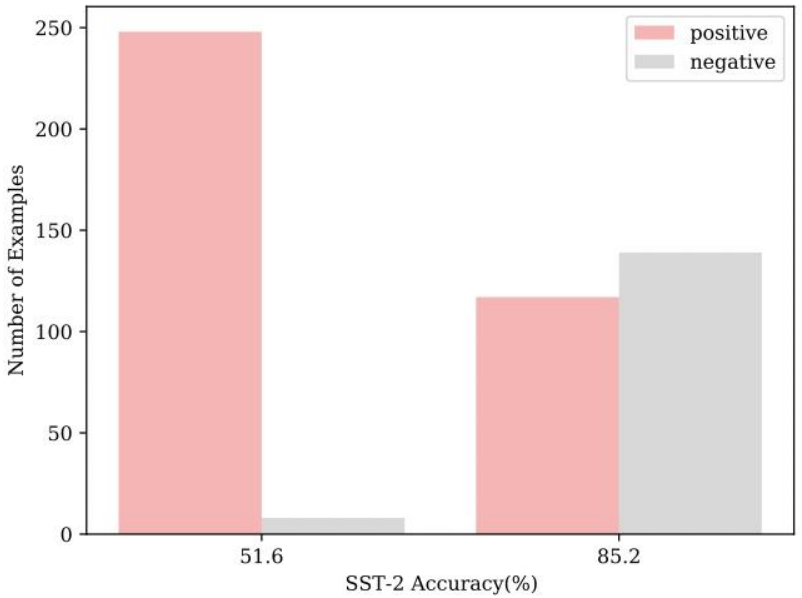


Although beneficial, increasing model size does not guarantee low variance

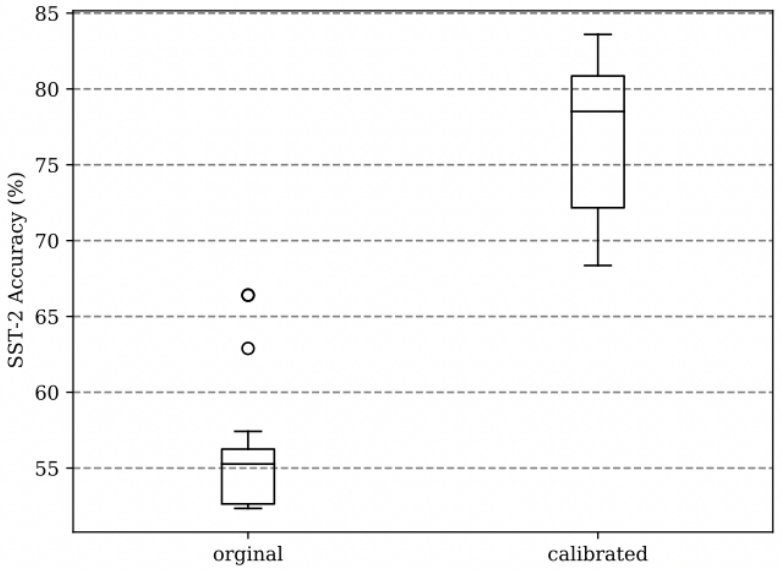
Prompt Design Study



Adding training samples does not significantly reduce variance



Failing prompts suffer from unbalanced label distribution

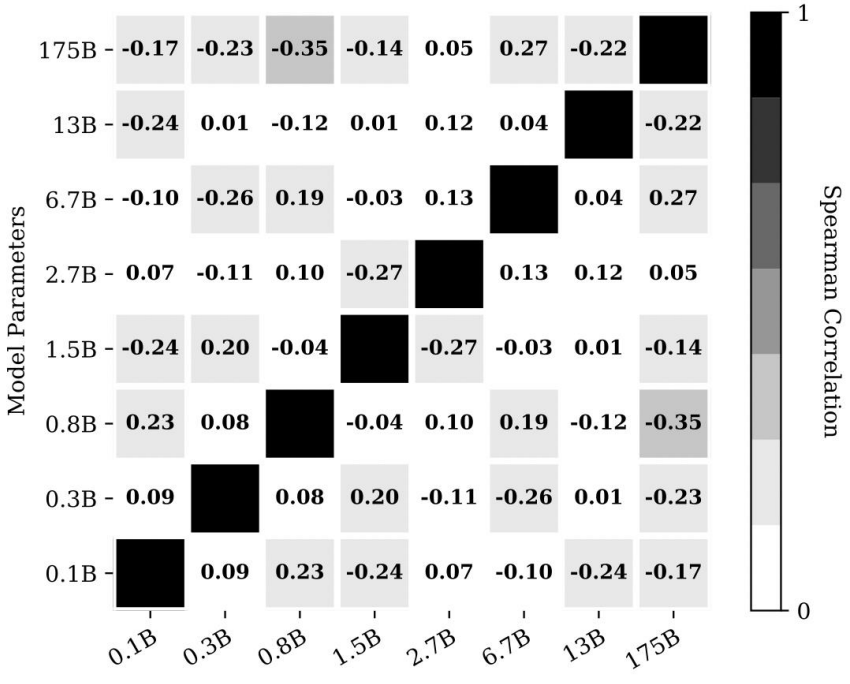


Calibration[1] improves performance but variance stays high

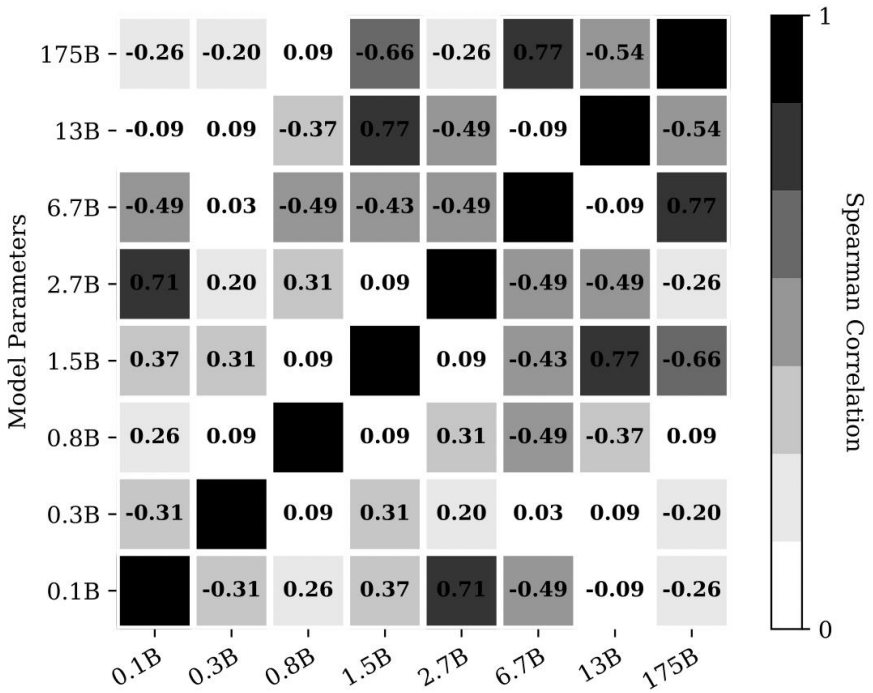
[1] Zhao et. al. Calibrate before use: Improving few-shot performance of language models. arXiv preprint arXiv:2102.09690.

✍: Aayush Mishra and Lingfeng Shen

Prompt Design Study



Performance Prompts are not transferable across models



Performant label orderings are not consistent across models

NNPP , NPNP , NPPN, PNNP , PNPN, PPNN

✍: Aayush Mishra and Lingfeng Shen

Spearman's rank correlation coefficient

The Spearman correlation coefficient is defined as the Pearson correlation coefficient between the rank variables.

$$r_s = \rho_{\mathrm{R}(X),\mathrm{R}(Y)} = \frac{\operatorname{cov}(\mathrm{R}(X), \mathrm{R}(Y))}{\sigma_{\mathrm{R}(X)} \sigma_{\mathrm{R}(Y)}},$$

$$\rho_{X,Y} = \frac{\operatorname{cov}(X,Y)}{\sigma_X \sigma_Y} \qquad \operatorname{cov}(X,Y) = \mathrm{E}\left[(X - \mathrm{E}[X])(Y - \mathrm{E}[Y])\right]$$

Only if all *n* ranks are *distinct integers*, it can be computed using the popular formula

$$r_s = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)},$$

where

$d_i = \mathrm{R}(X_i) - \mathrm{R}(Y_i)$ is the difference between the two ranks of each observation,

n is the number of observations.

IQ, X_i	Hours of TV per week, Y_i
106	7
100	27
86	2
101	50
99	28
103	29
97	20
113	12
112	6
110	17

IQ, X_i	Hours of TV per week, Y_i	rank x_i	rank y_i	d_i	d_i^2
86	2	1	1	0	0
97	20	2	6	−4	16
99	28	3	8	−5	25
100	27	4	7	−3	9
101	50	5	10	−5	25
103	29	6	9	−3	9
106	7	7	3	4	16
110	17	8	5	3	9
112	6	9	2	7	49
113	12	10	4	6	36

Prompt Engineering

How to find performant prompt orderings?

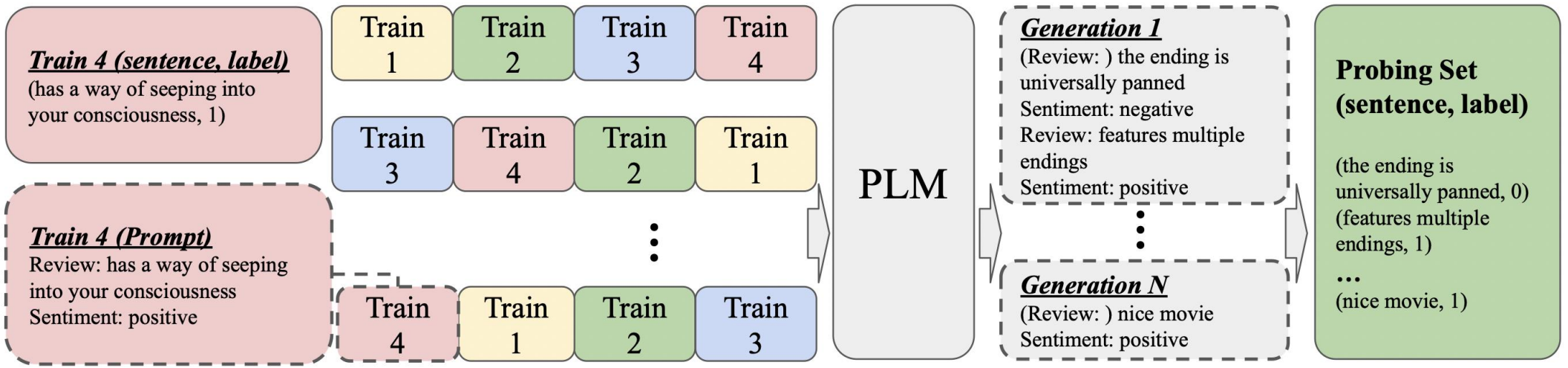
1. development set (need additional data)

2. automatically generate a ‘probing set’ (this paper)
(i) for a randomly-selected set of training samples, we use every possible ordering permutation of this set as candidates;
(ii) construct-ing a probing set by querying the language model using all candidate prompts as context;
(iii) use this probing set to identify the best ordering by ranking them using a probing metric.

✍: Aayush Mishra and Lingfeng Shen

Prompt Engineering

(ii) construct-ing a probing set by querying the language model using all candidate prompts as context



✍: Aayush Mishra and Lingfeng Shen

Probing Metrics

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

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

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

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

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

For prompts that avoid extremely unbalanced predictions.

To penalize overconfident predictions.

✍: Aayush Mishra and Lingfeng Shen

Experiments

- Models: GPT-2 (with 0.1B, 0.3B, 0.8B, and 1.5B parameters), GPT-3 (with 2.7B, and 175B parameters)
- 5 different sets of randomly selected training samples; 4 training smaples, 24 different permutations for each set
- select top k=4 samples ranked by highest entropy values
- classification datasets: SST-2, SST-5, DBPedia, MR, CR, MPQA, Subj, TREC, AGNews, RTE, CB
- Baselines:
  - Majority: predict the majority label in the dataset (lower-bound)
  - Oracle: select the top four orderings based on performance on the dev set (upper-bound)

https://www.gwern.net/GPT-3

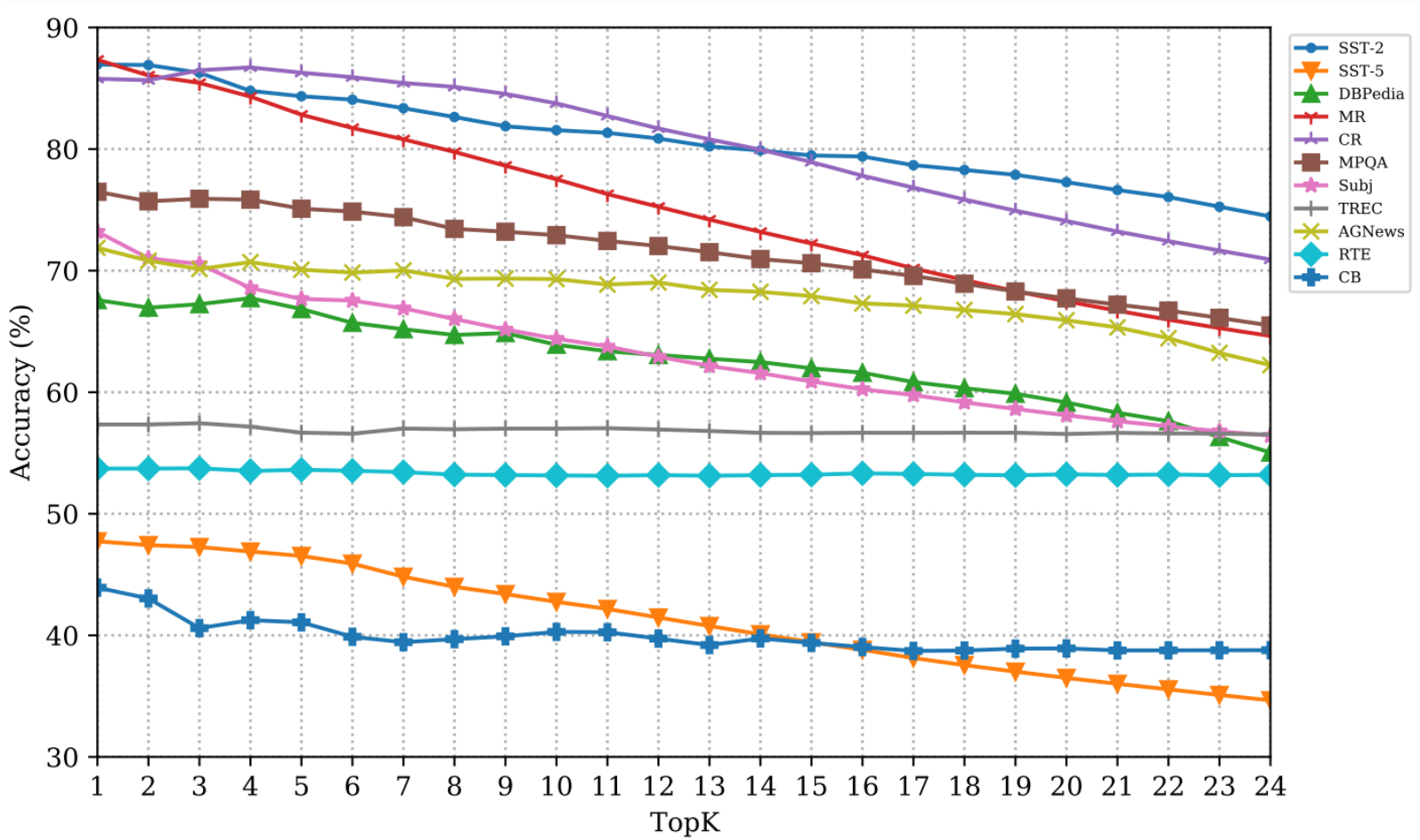
✍: Aayush Mishra and Lingfeng Shen

	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	$96.7_{0.2}$	$95.5_{0.2}$	$92.6_{0.4}$	$85.5_{0.8}$	$81.1_{4.9}$	$77.0_{1.2}$	$87.7_{0.6}$	$74.7_{0.4}$	$83.0_{0.9}$

✍: Aayush Mishra and Lingfeng Shen

Results

- Ranking using Entropy-based probing is robust



✍: Aayush Mishra and Lingfeng Shen

Results

- Entropy-based probing is effective across templates

	Template 1	Template 2	Template 3	Template 4
GPT-2 0.1B	$58.9_{7.8}$	$57.5_{6.8}$	$58.1_{7.4}$	$56.6_{6.6}$
LocalE	$\mathbf{65.2}_{3.9}$	$\mathbf{60.7}_{4.6}$	$\mathbf{65.4}_{4.8}$	$61.0_{4.7}$
GlobalE	$63.8_{5.8}$	$59.0_{2.9}$	$64.3_{4.8}$	$\mathbf{63.5}_{4.8}$
GPT-2 0.3B	$61.0_{13.2}$	$63.9_{11.3}$	$68.3_{11.8}$	$59.2_{6.4}$
LocalE	$75.3_{4.6}$	$70.0_{7.2}$	$80.2_{4.2}$	$62.2_{3.4}$
GlobalE	$\mathbf{78.7}_{5.2}$	$\mathbf{73.3}_{4.5}$	$\mathbf{81.3}_{4.1}$	$\mathbf{62.8}_{4.3}$
GPT-2 0.8B	$74.5_{10.3}$	$66.6_{10.6}$	$70.3_{10.5}$	$63.7_{8.9}$
LocalE	$81.1_{5.5}$	$80.0_{5.6}$	$73.7_{6.2}$	$\mathbf{71.3}_{4.5}$
GlobalE	$\mathbf{84.8}_{4.1}$	$\mathbf{80.9}_{3.6}$	$\mathbf{79.8}_{3.9}$	$70.7_{5.3}$
GPT-2 1.5B	$66.8_{10.8}$	$80.4_{7.6}$	$54.5_{7.9}$	$69.1_{10.5}$
LocalE	$76.7_{8.2}$	$83.1_{3.6}$	$66.9_{7.5}$	$72.7_{5.5}$
GlobalE	$\mathbf{81.8}_{3.9}$	$\mathbf{83.4}_{3.2}$	$\mathbf{67.2}_{6.1}$	$\mathbf{74.2}_{5.3}$

Prompt selection performance of different templates on SST-2

ID	Template	Label Mapping
1	Review: {Sentence} Sentiment: {Label}	positive/negative
2	Input: {Sentence} Prediction: {Label}	positive/negative
3	Review: {Sentence} Sentiment: {Label}	good/bad
4	{Sentence} It was {Label}	good/bad

Table 4: Different Templates for SST-2

✍: Aayush Mishra and Lingfeng Shen

Future directions

- Linguistic perspective
  - Are there any linguistic commonalities in these good orders?
  - How do these good orders arise?
  - Does it correlate with some linguistic distributions in the pre-trained corpus?

- Mathematical perspective
  - Does the uncertainty issue really come from biased/over-confident predictions?
  - Where does the uncertainty come from? (Error of estimated distribution towards ground-truth)
  - PAC-bayes or something?

✍: Aayush Mishra and Lingfeng Shen

Cons - Vicky

1. Unexplored theoretical grounding | Lack of transferability
    a. Prompt ordering affects performance greatly but is not transferable
        i. Why does ordering matter? Why is it not transferable? Is this similar to brute-force
    b. Probing metrics: Each motivation explained, but does not explain why only these two / how these two compare, and reason about their differing performances
2. Ablations not fully covered
    a. Argument on template invariance: Singled out sentiment analysis that inherently has limited template formats
    b. Lack of coverage on the 11 tasks evaluated: Pointed out sentence-pair tasks, but what about others? Complete breakdown beneficial
    c. Argument on probing to be better than train-devel split: Is it really better than original data, or is the split unfair? (Train set cut to half, expected drop)
3. General comments
    a. Figure captions can be improved
        i. Fig 1: Lack of description on variation within single sample run
        ii. Fig 3: Insufficient description on variance shade
        iii. Fig 4/5: Insufficient description on correlation value (small = worse)
    b. Introduce some context earlier for better grounding
        i. Reason for choosing 4-shot (limited by window size)
        ii. Each sample run is averaged across 5 subsets, each with 24 permutations

🎾 **Vicky Zeng and Aowei Ding**

On the Effect of Pretraining Corpora on In-context Learning by a Large-scale Language Model

Seongjin Shin[*,1] **Sang-Woo Lee**[*,1,2] **Hwijeen Ahn**[1] **Sungdong Kim**[2]
HyoungSeok Kim[1] **Boseop Kim**[1] **Kyunghyun Cho**[3] **Gichang Lee**[1]
Woomyoung Park[1] **Jung-Woo Ha**[1,2] **Nako Sung**[1]

NAVER CLOVA[1] NAVER AI Lab[2] NYU[3]

Pretraining corpus： HyperCLOVA

Name	Description	Tokens
Blog	Blog corpus	273.6B
Cafe	Online community corpus	83.3B
News	News corpus	73.8B
Comments	Crawled comments corpus	41.1B
KiN	Korean QnA website corpus	27.3B
Modu	Collection of five datasets	6.0B
Ency	Encyclopedia corpus	1.7B
Others	Other corpus	55.0B
Total		561.8B

Table 1: Descriptions of HyperCLOVA corpus (Kim et al., 2021).

max training set size：150B tokens

validation set size：10000 examples

Downstream corpus： HyperCLOVA

Name	Description
NSMC	a binary sentiment classification dataset on movie review
KorQuAD	a machine reading comprehension dataset similar to SQuAD 1.0
AI Hub translation	Korean-English parallel sentences from news, government websites, legal documents, etc
YNAT	a topic classification problem with seven classes

Relationship between pretraining corpus and downstream task

KorQuAD——Ency

YMAT——News

AI Hub——News

AI Hub——KiN

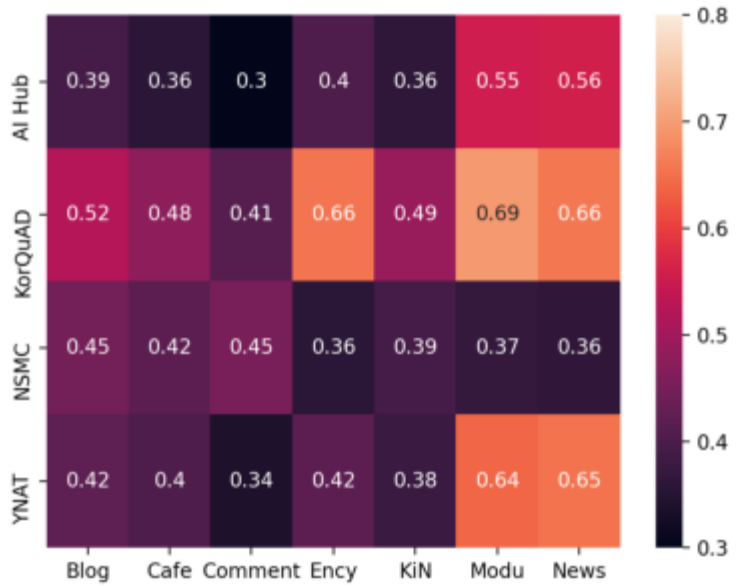


Figure 1: Vocabulary overlap ratio between pretraining corpus and downstream task. Top 1,000 nouns are used to calculate the ratio. Nouns are extracted using our in-house part-of-speech tagger.

Experiment

- 1. How large do the source and the size of pretraining corpora have the effects on emerging in-context learning ability?
- 2. What is the effect of combining various corpora?
- 3. How large does domain relevance of corpus influence on model performances of the downstream task?
- 4. How strong is the correlation between validation perplexity and in-context learning of language models?

Effect of Corpus Source

Model	Corpus Train	PPL	NSMC (Acc)	KorQuAD (EM)	KorQuAD (F1)	AI Hub (BLEU) Ko→En	AI Hub (BLEU) En→Ko	YNAT (F1)
Majority	-	-	50.35	0.0	0.0	0.0	0.0	8.26
ALL	150B	*119.99*	84.59	56.17	73.47	6.15	23.36	59.57
ALL w/o Others	150B	*119.66*	84.59	56.49	74.20	6.14	23.21	50.76
Blog	150B	152.40	**83.50**	**50.74**	**69.34**	**3.82**	**20.11**	**60.68**
Cafe	82.5B	170.85	57.77	3.12	14.26	2.83	**16.53**	11.04
News	73.1B	234.78	50.72	0.14	9.96	1.10	**15.88**	14.36
Comments	40.7B	225.39	**79.78**	14.69	33.33	0.79	5.06	**36.17**
KiN	27.0B	187.80	54.73	4.85	18.99	**6.81**	**18.16**	9.23
Modu	5.9B	226.01	**69.91**	**30.20**	**49.29**	1.21	6.13	**43.27**
Ency	1.7B	549.40	53.81	0.71	11.88	0.58	0.69	27.99
Blog 54B	54.0B	155.69	**83.06**	**49.13**	**68.10**	**3.93**	**21.12**	**57.97**
Blog 27B	27.0B	165.60	**80.27**	10.91	23.41	**5.35**	**12.32**	**48.19**
Cafe 27B	27.0B	169.81	49.91	1.37	13.98	**4.25**	**20.74**	8.60
News 27B	27.0B	239.79	50.64	0.80	8.02	2.42	**15.78**	27.20
Comments 27B	27.0B	229.65	**80.50**	13.02	31.53	1.70	3.28	25.79

In-context few-shot learning performance with different pretraining corpus

Effect of Corpus Size

# of tokens	NSMC (Acc)	KorQuAD (EM)	AI Hub (BLEU) Ko→En	AI Hub (BLEU) En→Ko	YNAT (F1)
150B	84.59	56.17	6.15	23.36	59.57
56B	84.35	55.13	5.47	22.98	51.89
6B	74.70	36.72	3.97	17.81	30.24

Table 6: In-context few-shot learning performance of ALL with different size of the pretraining data. The dataset is randomly sampled from the original corpus.

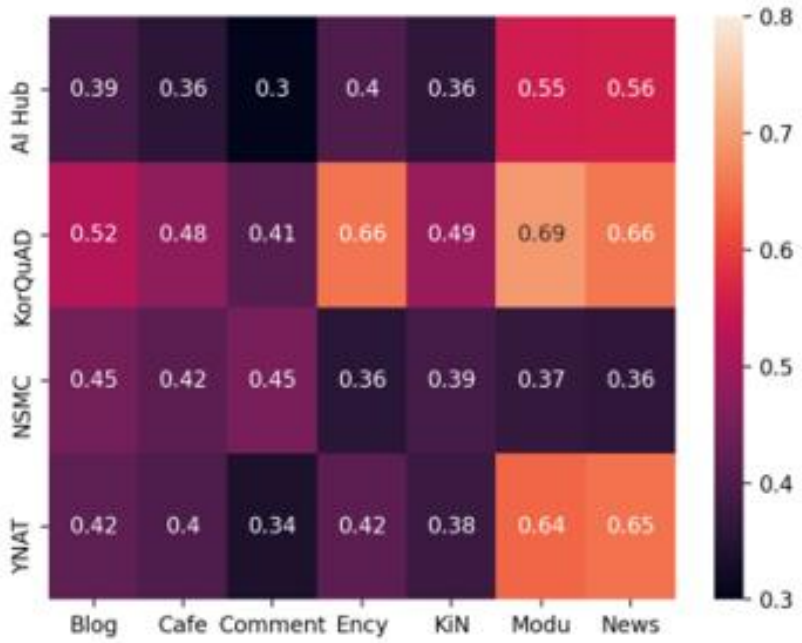
Effect of Combining Corpora

Corpus Type	Corpus Train	PPL	NSMC (Acc)	KorQuAD (EM)	KorQuAD (F1)	AI Hub (BLEU) Ko→En	AI Hub (BLEU) En→Ko	YNAT (F1)
ALL	150B	*119.99*	84.59	56.17	73.47	6.15	23.36	59.57
The Case where In-context few-shot learning Emerges by Combining Two Poor Corpora								
KiN+Ency	28.7B	*164.69*	59.17	**42.09**	**61.00**	**8.99**	**23.12**	**42.84**
Cafe+KiN	109.5B	*141.92*	**76.42**	**38.45**	**59.00**	**8.41**	**23.41**	**56.96**
The Case where In-context few-shot learning Does Not Emerge by Combining Two Poor Corpora								
Cafe+News	150B	*154.20*	54.15	8.95	22.72	**4.45**	**17.77**	8.19
The Case of Combining In-context few-shot Emerging Corpora								
Blog+Comments+Modu	150B	*144.67*	**82.82**	**54.94**	**72.27**	**4.09**	**21.17**	**65.01**
The Case of Adding News into KiN+Ency to Try to Enhance the Performance of YNAT								
News+KiN+Ency	101.8B	*142.13*	**75.96**	**35.42**	**55.60**	**8.70**	**23.38**	27.54

Table 4: In-context few-shot learning performance with different corpus combination.

Effect of Domain Relevance: few-shot

The close relationship between a pretraining corpus and a downstream task does not always guarantee in-context few-shot learning ability on the downstream task.



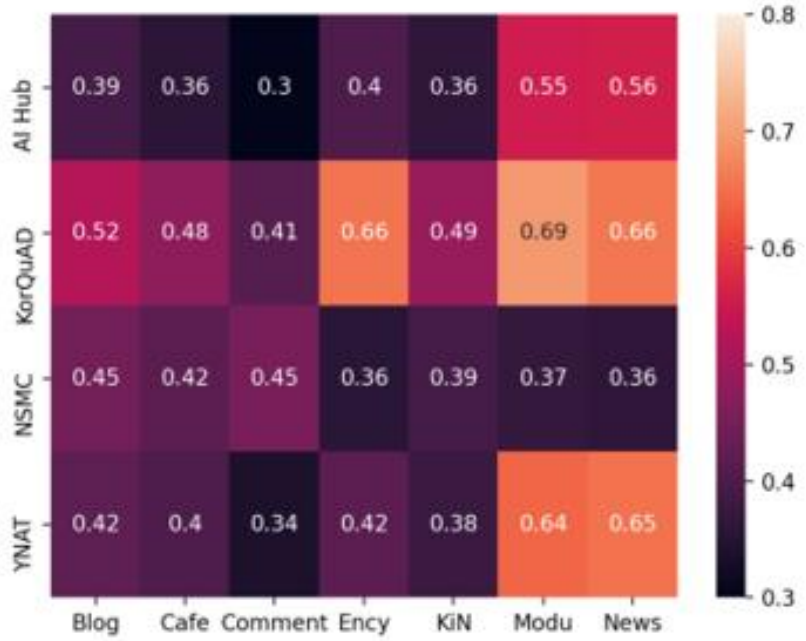
Model	Corpus Train	PPL	NSMC (Acc)	KorQuAD (EM)	KorQuAD (F1)	AI Hub (BLEU) Ko→En	AI Hub (BLEU) En→Ko	YNAT (F1)
Majority	-	-	50.35	0.0	0.0	0.0	0.0	8.26
ALL	150B	*119.99*	84.59	56.17	73.47	6.15	23.36	59.57
ALL w/o Others	150B	*119.66*	84.59	56.49	74.20	6.14	23.21	50.76
Blog	150B	152.40	**83.50**	**50.74**	**69.34**	**3.82**	**20.11**	**60.68**
Cafe	82.5B	170.85	57.77	3.12	14.26	2.83	**16.53**	11.04
News	73.1B	234.78	50.72	0.14	9.96	1.10	**15.88**	14.36
Comments	40.7B	225.39	**79.78**	14.69	33.33	0.79	5.06	**36.17**
KiN	27.0B	187.80	54.73	4.85	18.99	**6.81**	**18.16**	9.23
Modu	5.9B	226.01	**69.91**	**30.20**	**49.29**	1.21	6.13	**43.27**
Ency	1.7B	549.40	53.81	0.71	11.88	0.58	0.69	27.99
Blog 54B	54.0B	155.69	**83.06**	**49.13**	**68.10**	**3.93**	**21.12**	**57.97**
Blog 27B	27.0B	165.60	**80.27**	10.91	23.41	**5.35**	**12.32**	**48.19**
Cafe 27B	27.0B	169.81	49.91	1.37	13.98	**4.25**	**20.74**	8.60
News 27B	27.0B	239.79	50.64	0.80	8.02	2.42	**15.78**	27.20
Comments 27B	27.0B	229.65	**80.50**	13.02	31.53	1.70	3.28	25.79

Corpus Type	Corpus Train	PPL	NSMC (Acc)	KorQuAD (EM)	KorQuAD (F1)	AI Hub (BLEU) Ko→En	AI Hub (BLEU) En→Ko	YNAT (F1)
ALL	150B	*119.99*	84.59	56.17	73.47	6.15	23.36	59.57
The Case where In-context few-shot learning Emerges by Combining Two Poor Corpora								
KiN+Ency	28.7B	*164.69*	59.17	**42.09**	**61.00**	**8.99**	**23.12**	**42.84**
Cafe+KiN	109.5B	*141.92*	**76.42**	**38.45**	**59.00**	**8.41**	**23.41**	**56.96**
The Case where In-context few-shot learning Does Not Emerge by Combining Two Poor Corpora								
Cafe+News	150B	*154.20*	54.15	8.95	22.72	**4.45**	**17.77**	8.19
The Case of Combining In-context few-shot Emerging Corpora								
Blog+Comments+Modu	150B	*144.67*	**82.82**	**54.94**	**72.27**	**4.09**	**21.17**	**65.01**
The Case of Adding News into KiN+Ency to Try to Enhance the Performance of YNAT								
News+KiN+Ency	101.8B	*142.13*	**75.96**	**35.42**	**55.60**	**8.70**	**23.38**	27.54

Table 4: In-context few-shot learning performance with different corpus combination.

Effect of Domain Relevance: zero-shot

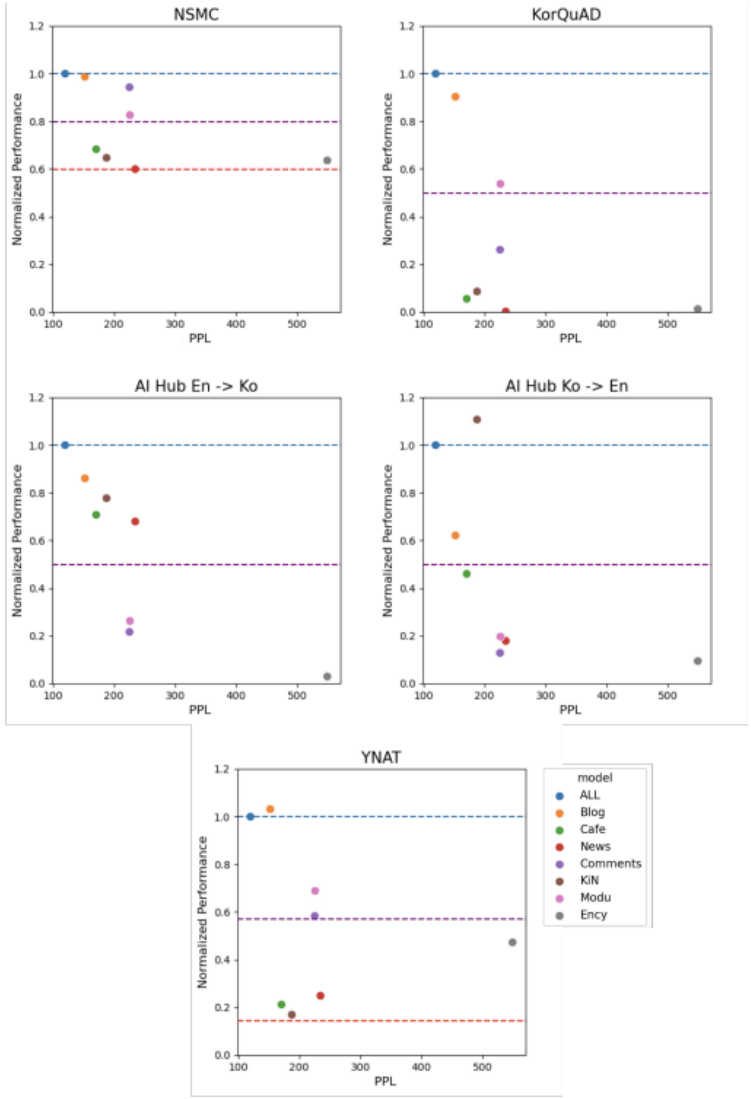
Domain relevance seems to affect more positively



Model	Corpus Train	PPL	NSMC (Acc)	KorQuAD (EM)	KorQuAD (F1)	AI Hub (BLEU) Ko→En	AI Hub (BLEU) En→Ko	YNAT (F1)
Majority	-	-	50.35	0.0	0.0	0.0	0.0	8.26
ALL	150B	*119.99*	61.68	**56.17**	**73.47**	**7.43**	**24.81**	**42.79**
Blog	150B	152.40	**75.28**	**50.74**	**69.34**	**5.44**	**22.88**	**49.34**
Cafe	82.5B	170.85	**69.38**	3.12	14.26	**4.34**	**16.44**	**38.12**
News	73.1B	234.78	54.96	0.14	9.96	1.28	10.21	**48.03**
Comments	40.7B	225.39	57.69	14.69	33.33	1.98	3.94	32.48
KiN	27.0B	187.80	65.43	4.85	18.99	**4.64**	10.42	**36.06**
Modu	5.9B	226.01	**72.50**	**30.22**	**49.30**	2.39	7.55	**35.28**
Ency	1.7B	549.40	42.96	14.01	31.51	0.80	0.77	30.22

Corpus Type	Corpus Train	PPL	NSMC (Acc)	KorQuAD (EM)	KorQuAD (F1)	AI Hub (BLEU) Ko→En	AI Hub (BLEU) En→Ko	YNAT (F1)
ALL	150B	*119.99*	61.88	**56.17**	**73.47**	**7.43**	**24.81**	**42.79**
KiN+Ency	28.7B	*164.69*	56.78	**42.09**	**61.00**	**11.51**	**24.93**	**37.71**
Cafe+KiN	109.5B	*141.92*	59.27	**38.45**	**59.00**	**10.12**	**24.95**	**45.44**
Cafe+News	150B	*154.20*	66.92	8.95	22.85	**3.49**	**15.77**	**47.34**
Blog+Comments+Modu	150B	*144.67*	**69.15**	**54.94**	**72.27**	**6.06**	**22.03**	**48.25**
News+KiN+Ency	101.8B	*142.13*	61.49	**35.42**	**55.60**	**10.18**	**24.13**	**51.89**

Perplexity and Downstream Task



In-context few-shot learning performance of various corpus models and their PPL.

Conclusion

- Corpus Source: In-context learning performance depends heavily on corpus sources, and with some sources, in-context learning does not work effectively.
- Corpus Combination: In-context learning ability can emerge by fusing two corpora, even when each on its own does not result in in-context learning.
- Domain Relevance: Pretraining with a corpus related to a downstream task seems to help in-context zero-shot learning performance, but is not indicative of the competitive in-context few-shot learning performance.
- Perplexity: Although perplexity and in-context learning accuracies correlate well when training a single model, perplexity alone does not reflect the difference in in-context learning accuracies across different language models.

Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?

Sewon Min[1,2] **Xinxi Lyu**[1] **Ari Holtzman**[1] **Mikel Artetxe**[2]
Mike Lewis[2] **Hannaneh Hajishirzi**[1,3] **Luke Zettlemoyer**[1,2]
[1]University of Washington [2]Meta AI [3]Allen Institute for AI
{sewon,alrope,ahai,hannaneh,lsz}@cs.washington.edu
{artetxe,mikelewis}@meta.com

- 1. The input-label mapping, i.e., whether each input xi is paired with a correct label yi.
- 2. The distribution of the input text, i.e., the underlying distribution that x1…xk are from.
- 3. The label space, i.e., the space covered by y1…yk.
- 4. The format—specifically, the use of input-label pairing as the format

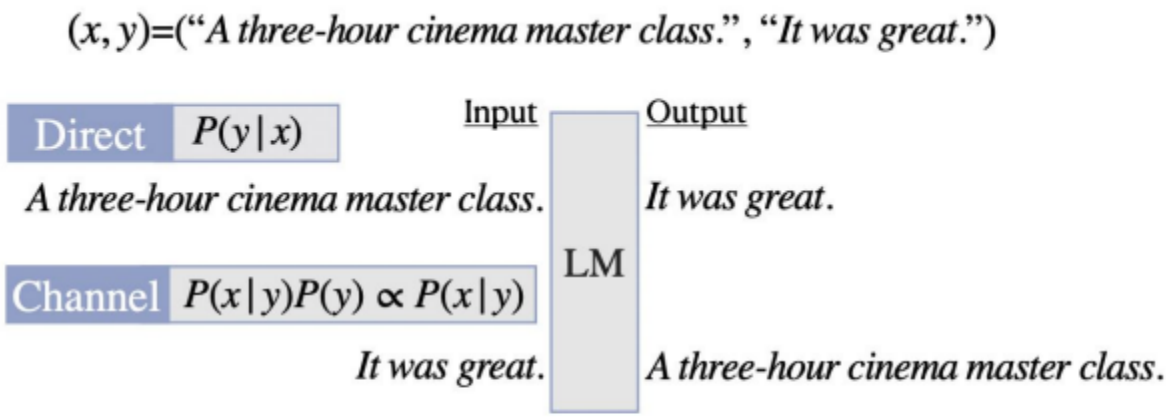
Model

Model	# Params	Public	Meta-trained
GPT-2 Large	774M	✓	✗
MetaICL	774M	✓	✓
GPT-J	6B	✓	✗
fairseq 6.7B†	6.7B	✓	✗
fairseq 13B†	13B	✓	✗
GPT-3	175B‡	✗	✗

MetaICL

	Meta-training	Inference
Task	C *meta-training* tasks	An unseen *target* task
Data given	Training examples $\mathcal{T}_i = \{(x_j^i, y_j^i)\}_{j=1}^{N_i},\ \forall i \in [1, C] \quad (N_i \gg k)$	Training examples $(x_1, y_1), \cdots, (x_k, y_k)$, Test input x
Objective	For each iteration, 1. Sample task $i \in [1, C]$ 2. Sample $k+1$ examples from $\mathcal{T}_i$: $(x_1, y_1), \cdots, (x_{k+1}, y_{k+1})$ 3. Maximize $P(y_{k+1}\|x_1, y_1, \cdots, x_k, y_k, x_{k+1})$	$\text{argmax}_{c \in \mathcal{C}} P(c\|x_1, y_1, \cdots, x_k, y_k, x)$

Direct vs Channel Models



P(y|x) ⟶ P(x|y)

Figure 1: An illustration of the direct model and the channel model for language model prompting in the sentiment analysis task.

Evaluation Data

Classification Tasks

Dataset	# Train	# Eval
Task category: Sentiment analysis		
financial_phrasebank	1,811	453
poem_sentiment	892	105
Task category: Paraphrase detection		
medical_questions_pairs	2,438	610
glue-mrpc	3,668	408
Task category: Natural language inference		
glue-wnli	635	71
climate_fever	1,228	307
glue-rte	2,490	277
superglue-cb	250	56
sick	4,439	495
Task category: Hate speech detection		
hate_speech18	8,562	2,141
ethos-national_origin	346	87
ethos-race	346	87
ethos-religion	346	87
tweet_eval-hate	8,993	999
tweet_eval-stance_atheism	461	52
tweet_eval-stance_feminist	597	67
Task category: Question answering		
quarel	1,941	278
openbookqa	4,957	500
qasc	8,134	926
commonsense_qa	9,741	1,221
ai2_arc	1,119	299
Task category: Sentence completion		
codah	1665	556
superglue-copa	400	100
dream	6116	2040
quartz-with_knowledge	2696	384
quartz-no_knowledge	2696	384

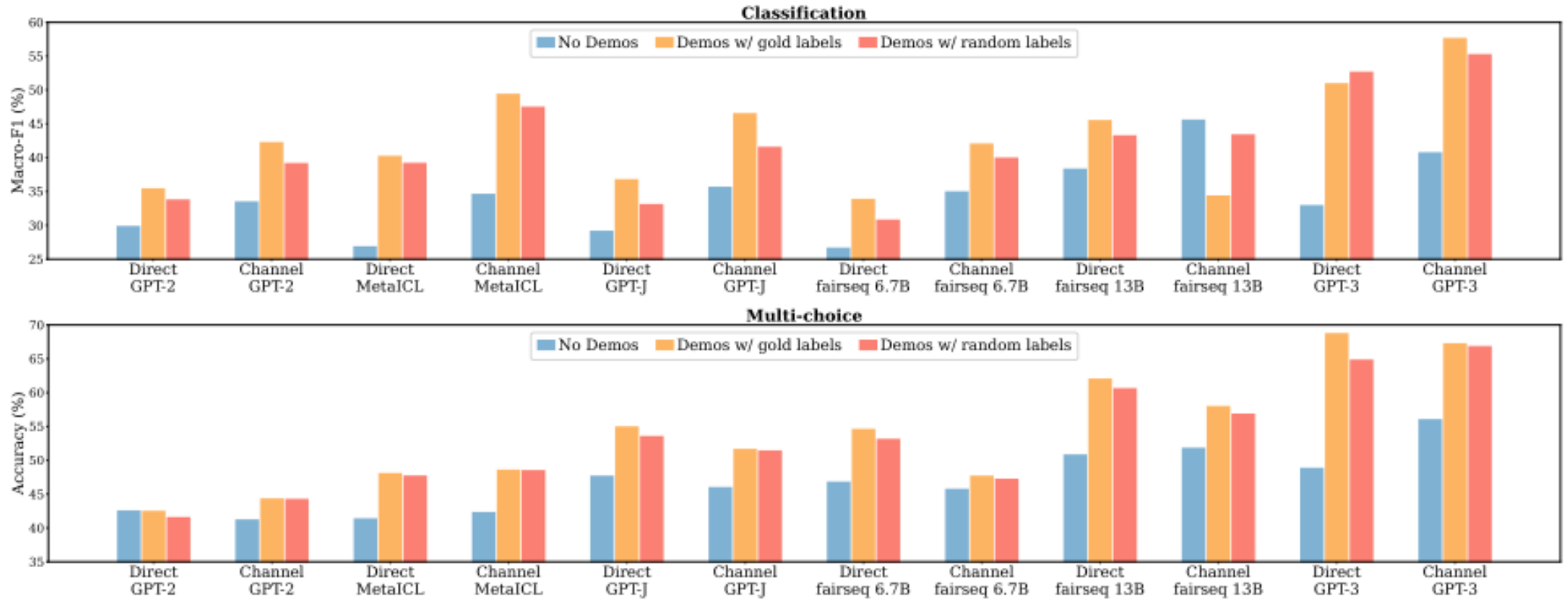
Multiple Choice Tasks

Some Details

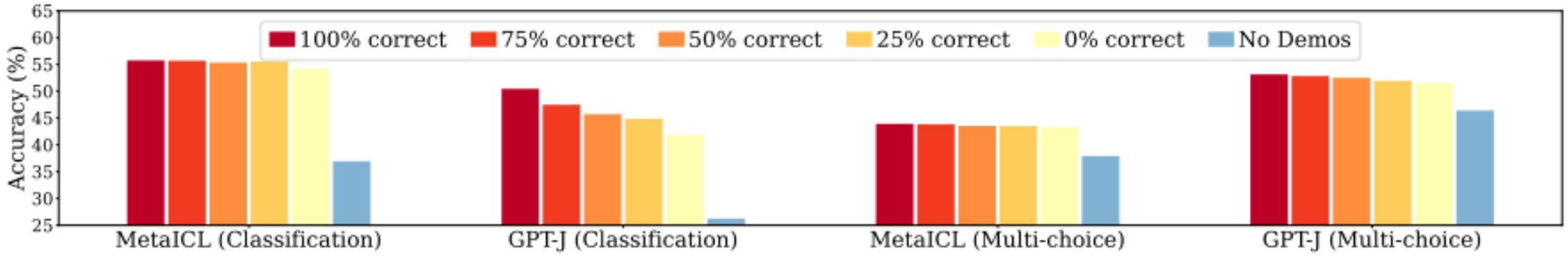
- k = 16 examples as demonstrations
- report Macro-F1 for classification tasks and Accuracy for multi-choice tasks
- minimal templates

Dataset	Type	Example
MRPC	Minimal	sentence 1: Cisco pared spending to compensate for sluggish sales . [SEP] sentence 2: In response to sluggish sales , Cisco pared spending . \n {equivalent|not_equivalent}
	Manual	Cisco pared spending to compensate for sluggish sales . \n The question is: In response to sluggish sales , Cisco pared spending . True or False? \n The answer is:{True|False}
RTE	Minimal	sentence 1: The girl was found in Drummondville. [SEP] sentence 2: Drummondville contains the girl. \n {entailment|not_entailment}
	Manual	The girl was found in Drummondville. \n The question is: Drummondville contains the girl. True or False? \n The answer is:{True|False}
Tweet_eval-hate	Minimal	The Truth about #Immigration \n {hate|non-hate}
	Manual	Tweet: The Truth about #Immigration \n Sentiment: {against|favor}
SICK	Minimal	sentence 1: A man is screaming. [SEP] sentence 2: A man is scared. \n {contradiction|entailment|neutral}
	Manual	A man is screaming. \n The question is: A man is scared. True or False? \n The answer is: {False|True|Not sure}
poem-sentiment	Minimal	willis sneered: \n {negative|no_impact|positive}
	Manual	willis sneered: \n The sentiment is: {negative|no_impact|positive}
OpenbookQA	Minimal	What creates a valley? \n {feet|rock|water|sand}
	Manual	The question is: What creates a valley? \n The answer is: {feet|rock|water|sand}
CommonsenseQA	Minimal	What blocks sunshine? \n {summer|park|desktop|sea|moon}
	Manual	The question is: What blocks sunshine? \n The answer is: {summer|park|desktop|sea|moon}
COPA	Minimal	Effect: I coughed. \n {Cause: I inhaled smoke.|Cause: I lowered my voice.}
	Manual	I coughed because {I inhaled smoke.|I lowered my voice.}
ARC	Minimal	Which biome has the most vegetation? \n {desert|forest|grassland|tundra}
	Manual	The question is: Which biome has the most vegetation? \n The answer is: {desert|forest|grassland|tundra}

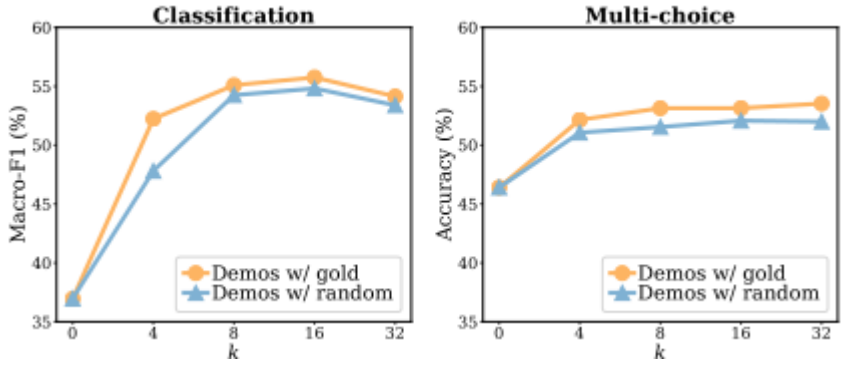
Experiment: Input-label Mapping

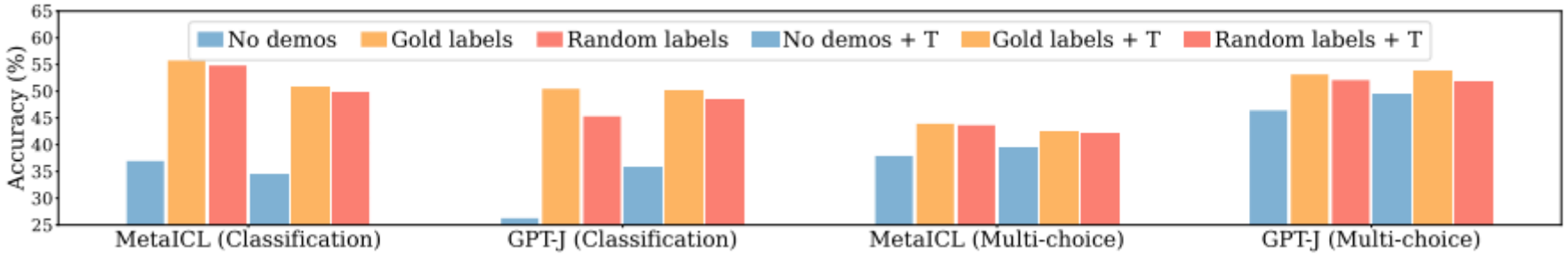


Experiment: Input-label Mapping

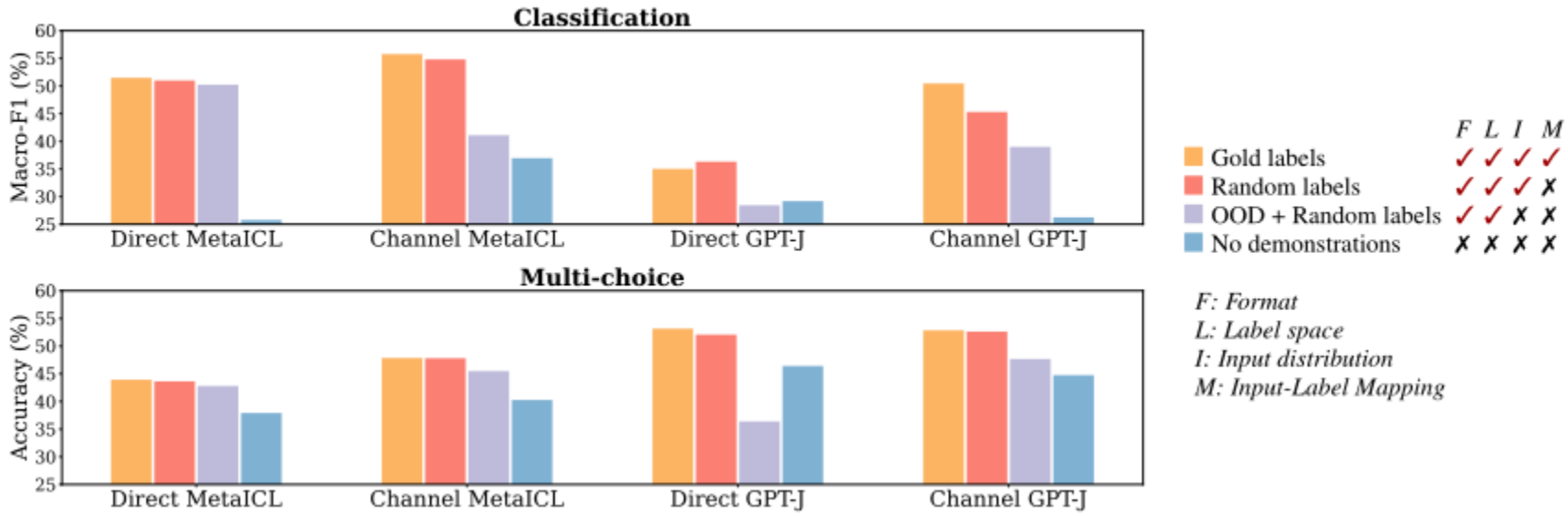


Experiment: Input-label Mapping

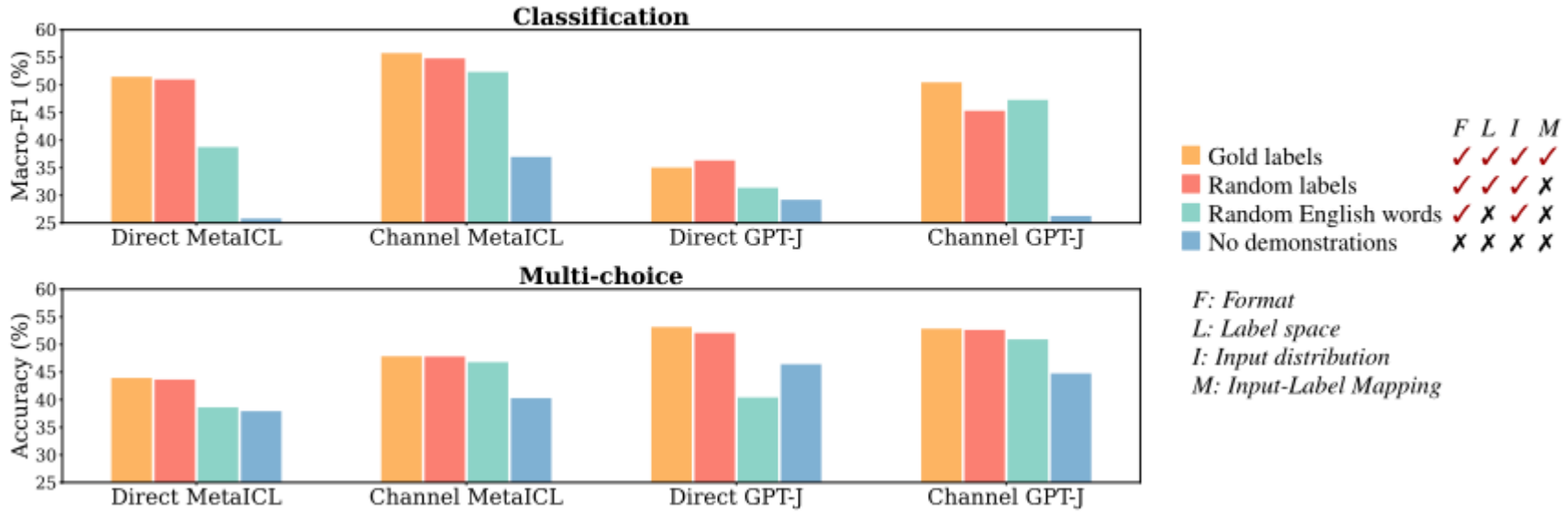




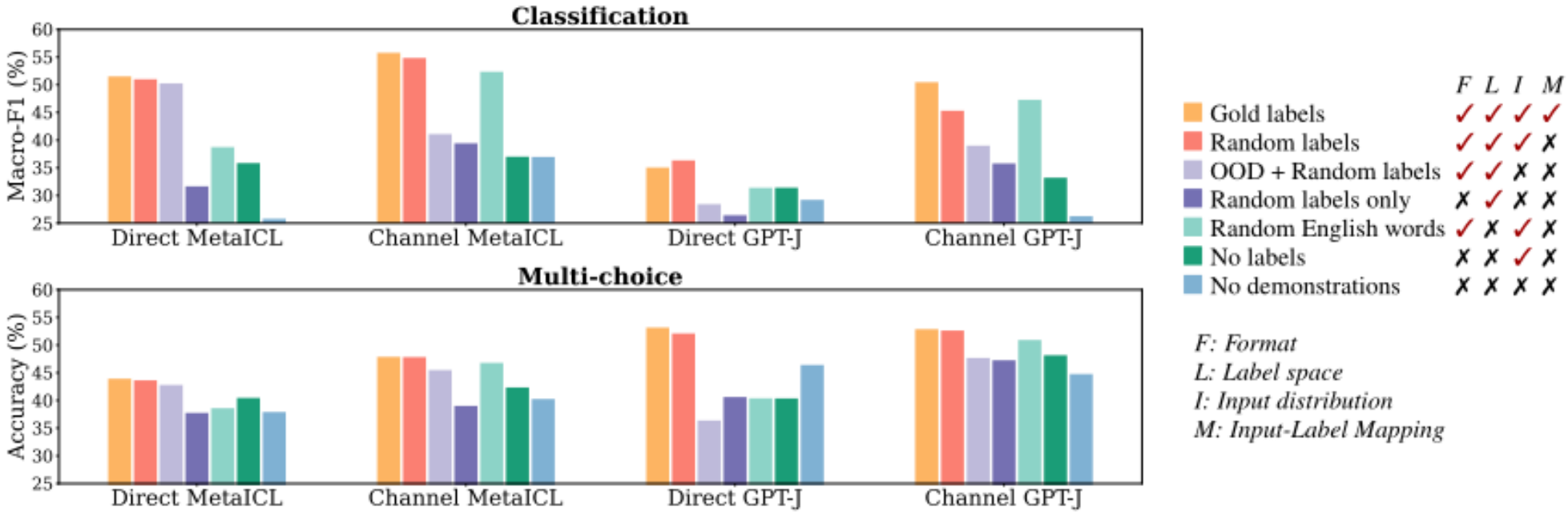
Experiment: Impact of the distribution of the input text



Experiment: Impact of the label space



Experiment: Impact of the format



Experiment: meta-training

the ground truth input-label mapping matters even less

keeping the format of the demonstrations matters even more

nearly zero influence of the input-label mapping and the input distribution in Direct MetaICL

nearly zero influence of the input-label mapping and the output space in Channel MetaICL

(1) the input-label mapping is likely harder to exploit,
(2) the format is likely easier to exploit, and
(3) the space of the text that the model is trained to generate is likely easier to exploit than the space of the text that the model conditions on.