Understanding In-Context Learning via Supportive Pretraining Data

Xiaochuang Han[♠♣*] **Dániel Simig**[♣] **Todor Mihaylov**[♣]
Yulia Tsvetkov[♠] **Asli Celikyilmaz**[♣] **Tianlu Wang**[♣]
[♣]Meta AI
[♠]University of Washington
{xhan77, yuliats}@cs.washington.edu simigd@gmail.com {tbmihaylov, aslic, tianluwang}@meta.com

Abstract

In-context learning (ICL) improves language models' performance on a variety of NLP tasks by simply demonstrating a handful of examples at inference time. It is not well understood why ICL ability emerges, as the model has never been specifically trained on such demonstrations. Unlike prior work that explores implicit mechanisms behind ICL, we study ICL via investigating the *pretraining data*. Specifically, we first adapt an iterative, gradient-based approach to find a small subset of pretraining data that *supports* ICL. We observe that a continued pretraining on this small subset significantly improves the model's ICL ability, by up to 18%. We then compare the supportive subset constrastively with random subsets of pretraining data and discover: (1) The supportive pretraining data to ICL do *not* have a higher domain relevance to downstream tasks. (2) The supportive pretraining data have a higher mass of rarely occurring, long-tail tokens. (3) The supportive pretraining data are *challenging* examples where the information gain from long-range context is below average, indicating learning to incorporate difficult long-range context encourages ICL. Our work takes a first step towards understanding ICL via analyzing instance-level pretraining data. Our insights have a potential to enhance the ICL ability of language models by actively guiding the construction of pretraining data in the future.

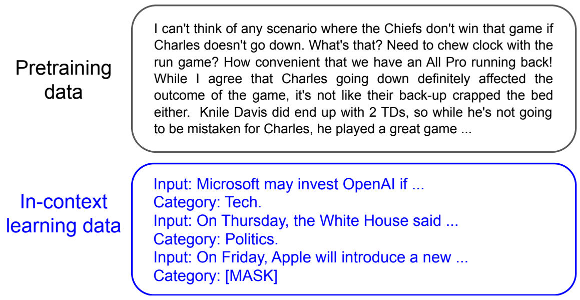

Figure 1: An example from the pretraining data of OPT (Zhang et al., 2022b) and an illustrative in-context learning example of topic classification. The in-context learning task data can be drastically different from pretraining instances, both in content and format.

1 Introduction

In-context learning in NLP has drawn tremendous attention recently (Dong et al., 2022). Unlike traditional learning paradigms that rely on training or finetuning models, in-context learning only provides a handful of demonstration examples to language models as a prefix to the test input, without any parameter updates. In-context learning has shown superior performance on a range of NLP tasks (Brown et al., 2020; Zhang et al., 2022b; Chowdhery et al., 2022; Hoffmann et al., 2022), but the origin and reason of this emergent ability remain under-investigated. In-context learning is surprising since language models have not been explicitly trained to learn from demonstration examples (Xie et al., 2022). As shown in an illustrative scenario in Figure 1, a typical pretraining data instance is highly different from an in-context learning example for downstream tasks, in both content and format.

Prior work have attempted to answer *what* in-context learning is, through empirically investigating useful and irrelevant attributes of the demonstration examples (Min et al., 2022; Zhang et al., 2022a), or theoretically proving certain synthetic language models implicitly do Bayesian inference with demonstrations (Xie et al., 2022). Furthermore, recent work have drawn connections between the mechanism of in-context learning and standard learning algorithms, such as regression, nearest neighbor, and gradient descent (Olsson et al., 2022; Akyürek et al., 2022; Dai et al., 2022; von Oswald et al., 2022).

Differently, in this work we are interested in understanding *from where* the in-context learning ability is acquired, through a perspective of pre-

*Work done during an internship at Meta AI.

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 12660–12673
July 9-14, 2023 ©2023 Association for Computational Linguistics

training data. Although not many, some recent work have investigated this direction. For instance, Shin et al. (2022) pretrain a variety of language models on different corpora. They study correlations between attributes of pretraining datasets and in-context learning performance, at a relatively coarse dataset-level. Chan et al. (2022) construct pretraining data with different attributes and discover that some distributional properties of the data drive the emergence of in-context learning. However, their experiment is limited to synthetic data of image-label pairs.

In this work, we investigate a large language model OPT (Zhang et al., 2022b) and its pretraining data. We first hypothesize that there exists some specific pretraining data instances that are particularly helpful to the model's in-context learning ability. As an attempt to find such instances, we adapt an iterative, gradient-based method ORCA (Han and Tsvetkov, 2022) to search within OPT's pretraining corpus. The process is guided by the gradients of the in-context learning data from downstream tasks, and we refer to the identified subset as supportive pretraining data to in-context learning following Han and Tsvetkov (2022). Furthermore, we quantitatively verify through a perturbative continued pretraining, that the supportive subset does improve the model's in-context learning performance on downstream tasks, while not affecting a spurious zero-shot performance (§2).

We then analyze the identified supportive data in contrast to the general pretraining data, to obtain data features particularly relevant to in-context learning. We specifically approach from three aspects: the domain relevance to downstream tasks, the token frequency distribution, and the information gain of incorporating long-range pretraining context. Our major findings include: **(1)** Compared to general pretraining data, the supportive data do *not* have a higher domain relevance to the downstream tasks. **(2)** The supportive pretraining data contain a relatively higher amount of rarely occurring, long-tail tokens. **(3)** The supportive pretraining data are *challenging* examples in incorporating long-range context for language modeling (§3).

Our work offers a first step towards interpreting in-context learning in NLP tasks via analyzing instance-level pretraining data. We believe it can help improve the transparency and interpretability of language models' in-context learning behavior. Our analysis can also pave the way to improved in-context learning in the future by informing pretraining data construction.

2 Finding supportive pretraining data for in-context learning

Han and Tsvetkov (2022) propose an iterative, gradient-based method ORCA to find supportive pretraining data of BERT (Devlin et al., 2019) under a vanilla zero-shot prompting setup. In this section, we provide some background and adapt ORCA for large language models in a setting of in-context learning (ICL), finding supportive pretraining data for downstream tasks with demonstration examples.[1]

2.1 Methodology

Assume we have a pretrained language model (LM) θ and data pairs $(\boldsymbol{x}, \boldsymbol{y})$ representing the inputs and ground truth outputs of task D_{task}. Both $\boldsymbol{x}$ and $\boldsymbol{y}$ are in natural language. For classification tasks, the target labels can be converted to natural language via verbalizers (Schick and Schütze, 2021).

Zero-shot prompting A pretrained language model can be applied to perform downstream tasks via zero-shot prompting (e.g., Petroni et al., 2019). For classification tasks, the language model θ outputs a candidate answer with top probability, $\operatorname{argmax}_{\boldsymbol{y}' \in \mathcal{Y}} p_\theta(\boldsymbol{y}' \mid \boldsymbol{x}) = \operatorname{argmax}_{\boldsymbol{y}' \in \mathcal{Y}} \prod_{t=0}^{t<|\boldsymbol{y}'|} p_\theta(y'_t \mid \boldsymbol{x}, \boldsymbol{y}'_{<t})$, where $\mathcal{Y}$ contains all candidate answers $\boldsymbol{y}'$. For generation tasks, outputs can be obtained by sampling autoregressively from θ conditioned on $\boldsymbol{x}$ (e.g., Holtzman et al., 2019). This is a zero-shot scenario with no demonstration examples.

In-context learning Instead of modeling $p_\theta(\boldsymbol{y} \mid \boldsymbol{x})$, ICL estimates $p_\theta(\boldsymbol{y} \mid \{(\boldsymbol{x}_{\text{demo}}, \boldsymbol{y}_{\text{demo}})\}, \boldsymbol{x})$, prepending the original model input with several demonstration examples $(\boldsymbol{x}_{\text{demo}}, \boldsymbol{y}_{\text{demo}})$ sampled from the target task D_{task}. The language model θ is never trained on the task data with demonstrations. However, we can form a loss on the in-context data as a surrogate for θ's

[1] Identifying important training data for an inference time model output is an estabilished topic in model interpretability, with various prior work measuring data importance via variants of gradient similarity (Koh and Liang, 2017; Pruthi et al., 2020). However, these methods are prohibitively expensive to be applied to large-scale pretraining data. Concurrent to our work, Guu et al. (2023) propose an interesting method to model the importance of individual training examples by simulating training runs, but it is also on a scale of finetuning instead of pretraining.

ICL performance, which will be used for a later guidance step, $L_\theta^{\text{ICL}}(\boldsymbol{x},\boldsymbol{y}) = -\log p_\theta(\boldsymbol{y} \mid \{(\boldsymbol{x}_{\text{demo}},\boldsymbol{y}_{\text{demo}})\},\boldsymbol{x}) = -\log \prod_{t=0}^{t<|\boldsymbol{y}|} p_\theta(y_t \mid \{(\boldsymbol{x}_{\text{demo}},\boldsymbol{y}_{\text{demo}})\},\boldsymbol{x},\boldsymbol{y}_{<t})$.

Pretraining The pretraining data of θ often consists of texts $\boldsymbol{w}$ from large, general-domain corpora. During pretraining, the LM θ is updated via stochastic gradient descent with a loss to reconstruct $\boldsymbol{w}$ given a prefixing context, $L_\theta^{\text{PT}}(\boldsymbol{w}) = -\log \prod_{t=0}^{t<|\boldsymbol{w}|} p_\theta(w_t \mid \boldsymbol{w}_{<t})$.

Supportive pretraining data Our goal is to locate what pretraining data $\boldsymbol{w}$ if upweighted would be most helpful to the LM θ's ICL ability. Following ORCA (Han and Tsvetkov, 2022), we use the similarity between gradients $\nabla_\theta L_\theta^{\text{PT}}(\boldsymbol{w})$ and $\nabla_\theta L_\theta^{\text{ICL}}(\boldsymbol{x},\boldsymbol{y})$ iteratively to find such supportive pretraining data. We show details of our adapted algorithm ORCA-ICL in Figure 2. The algorithm finds pretraining data that exert a gradient to θ similarly as a group of guidance ICL task data would. $\nabla_\theta L_\theta^{\text{ICL}}(\boldsymbol{x},\boldsymbol{y})$ provides a guidance for the direction the model parameters *should* be updated towards to be better at ICL, while $\nabla_\theta L_\theta^{\text{PT}}(\boldsymbol{w})$ approximates how the direction the model parameters *would* be updated based on individual pretraining instances. We conduct a multi-iteration process (a total of M iterations each selecting k supportive instances) to mitigate noise.[2] SGD denotes an one-pass stochastic gradient descent to mimick an incremental upweight to the selected data, with a minimum number of steps to prevent overfitting. The resulting supportive set S has a very small size (under 2000 in this work).[3]

Verifying supportiveness To quantitatively evaluate the supportiveness of the selected set of pretraining data, we perform an one-pass gradient descent on the original LM with the selected set S, which mimics a *perturbative continued pretraining* with a minimum number of updates: $\theta_M \leftarrow \underset{S}{\text{SGD}}(\theta_0)$. We then benchmark this perturbed model (θ_M) with the original model (θ_0) and a model perturbed with a random set of pretraining data. We expect the perturbed model using our selected supportive pretraining data to achieve a better ICL performance.

Algorithm 1 ORCA-ICL

Load a pretrained language model as θ_0
for $i \leftarrow 1, M$ **do**
  if $i = 1$ **then**
    $S_1 \leftarrow \underset{\boldsymbol{w}\in D_{\text{PT}}}{\text{argtop-}k}[\cos(\nabla_\theta L_{\theta_0}^{\text{PT}}(\boldsymbol{w}), \nabla_\theta \sum_{D_{\text{task}}} L_{\theta_0}^{\text{ICL}}(\boldsymbol{x},\boldsymbol{y}))]$
    $\theta_1 \leftarrow \underset{S_1}{\text{SGD}}(\theta_0)$
  else
    $S_i \leftarrow \underset{\boldsymbol{w}\in D_{\text{PT}}}{\text{argtop-}k}[\cos(\nabla_\theta L_{\theta_0}^{\text{PT}}(\boldsymbol{w}), \nabla_\theta \sum_{D_{\text{task}}} L_{\theta_{i-1}}^{\text{ICL}}(\boldsymbol{x},\boldsymbol{y}))]$
    $\theta_i \leftarrow \underset{\cup_{j=1}^{i} S_j}{\text{SGD}}(\theta_0)$
  end if
end for
Return supportive pretraining data $S \leftarrow \cup_{i=1}^{M} S_i$

Figure 2: ORCA-ICL, an iterative gradient-based selection of supportive pretraining data for ICL.

2.2 Setup

Language model Throughout the work, we use a pretrained, autoregressive OPT-6.7B (Zhang et al., 2022b) as our LM θ.

Tasks In this work, we focus on classification problems and first retrieve 48 classification-based tasks from Natural Instructions v2 (NI-v2, Wang et al., 2022). We apply the LM on the tasks with both a zero-shot and in-context learning setup. We extract tasks that achieve at least 10% better performance with in-context demonstrations. We group 17 tasks that satisfies the constraint and further select 6 typical tasks among them:

SST-2: Movie review sentiment classification (Socher et al., 2013). **AG News**: News topic classification (Zhang et al., 2015). **Story Cloze Test**: Story coherence classification (Mostafazadeh et al., 2017). **SMS Spam Collection**: Spam classification (Almeida et al., 2011). **Sentiment 140**: Tweet sentiment classification (Go et al., 2009). **TweetQA**: Answer verification (Xiong et al., 2019).

For each task, we randomly sample 500 examples with a balanced class distribution as D_{task}, guiding the ORCA-ICL algorithm. The quantitative evaluation is performed on the full dataset. For ICL, for each instance in the task data, we randomly sample 4 demonstration examples under each candidate class defined in the task.[4] The order of demonstration examples in the context is randomly shuffled. The template and verbalizer of each task follows the original NI-v2 dataset, though we did not include the task instructions, as

[2] Additionaly according to Han and Tsvetkov (2022), this may prevent selecting examples associated with only one class of the task, a case of poor calibration.

[3] More details of the ORCA algorithm can be found in Han and Tsvetkov (2022).

[4] The sampling of demonstration examples is independent across test instances to mitigate potential spurious correlations.

the focus of this work is in-context learning with demonstration examples.

Pretraining Considering the size of pretraining data D_{PT}, we include an as large portion of OPT's pretraining data as possible under a reasonable budget. Specifically, in this work we use a total of 2.5M pretraining instances each consists of 2048 tokens.[5] For computing efficiency, we use intra-layer model parallelism (Shoeybi et al., 2019) and fully sharded data parallel (Ott et al., 2021).[6]

Implementation Details We run ORCA-ICL with a maximum of $M = 5$ iterations. In each iteration we extract $k = 400$ pretraining instances with top gradient similarity with the ICL task data. We use a batch size of 16 and learning rate of 2e-5 for the one-pass gradient descent with an Adam optimizer (Kingma and Ba, 2014). This results in a total of 125 updates[7] to the original LM after all iterations as the perturbative continued pretraining.

2.3 Results

Perturbative continued pretraining As the main evaluation of the supportive pretraining data obtained by ORCA-ICL, we perform perturbative continued pretraining on both the selected supportive data and random pretraining data as a control. Table 1 shows the main results of task accuracy. The leftmost column shows a source task D_{task} guiding the selection of supportive pretraining data. At each row, we evaluate the perturbed model ($\underset{S}{\text{SGD}}(\theta_0)$) on all 6 tasks. The ICL performance of the original LM is reported in the headers of the table.

In each cell of the table, the top number shows the continued pretraining result with the supportive data we identified. We consider $M \in [1, 5]$ iterations as a hyperparameter and report result with a best M. We want to know *at a same size of selection*, how our identified subset performs compared to random pretraining data. We therefore run random selection with 5 seeds, and the bottom number of the cell shows the continued pretraining result with random data at a same size of our selection, accompanied by a standard deviation. The performance of our selection is bolded when the performance difference with random selection exceeds one standard deviation.

The diagonal cells show the performance of perturbed models on the same task used for selecting supportive data. We observe on 4 of the 6 source tasks, our selection of supportive pretraining data is effective. For the cross-task performance, we observe on 5 of the 6 source tasks, our selection is effective for at least three tasks.[8] We conclude that **our identified supportive pretraining data is overall effective for ICL**, though the cross-task results show a portion of the ICL behavior can be task-specific and not universal across tasks.

Control evaluation on zero-shot data Being effective on the ICL data does not necessarily mean a direct support for a model's ICL ability, which is to learn from the demonstration examples. The test input can be a confounding factor: if our selection is effective as well on zero-shot test input without demonstrations, then the selection is not specific to the ICL ability. Therefore, we further confirm the supportiveness of our selected supportive pretraining data to ICL, contrastively in a zero-shot setup. We evaluate our models after perturbative continued pretraining in Table 1 on the same tasks but without the in-context demonstrations. We present the results in Table 2. The two columns show the zero-shot prompting performance of the original LM and the model after continued pretraining with our ICL-supportive selection, respectively. We do not observe performance gain for most tasks, indicating **our selection is specific to the ICL ability without benefiting the zero-shot, no-demonstration task performance**.

3 Analyzing supportive pretraining data for in-context learning

In the previous section, we identify a small subset of pretraining data that supports the ICL ability of language models. In this section, we analyze the selected supportive pretraining data to understand what makes them useful to ICL. Specifically, we compare the supportive pretraining data contrastively with randomly sampled pretraining instances, investigating three aspects of the pretraining data: the domain relevance to downstream

[5]The total 5B tokens are about 3% of OPT's 180B full pretraining data.

[6]This groups 4 input data for each backward pass in our setup. The 4 instances receive a same gradient similarity score, equivalent to an aggregated instance 4 times of the length.

[7]The one-pass descent has $\frac{M*k}{\text{batch size}}$ steps.

[8]Negative result is observed with TweetQA, on which we conjecture the patterns in the demonstration examples are more difficult to transfer to the test input (e.g., factual knowledge instead of sentiment indicators).

Eval / *Source*	SST-2 75.47	AG News 74.12	Story Cloze 66.09	SMS Spam 45.07	Sentiment 140 67.23	TweetQA 62.36
SST-2	**83.15** 75.87$_{\pm 1.64}$	**74.91** 73.24$_{\pm 1.24}$	**67.76** 66.24$_{\pm 1.25}$	52.48 49.82$_{\pm 4.50}$	**69.03** 66.23$_{\pm 1.24}$	**62.20** 61.75$_{\pm 0.26}$
AG News	**79.04** 74.99$_{\pm 0.77}$	**75.40** 73.77$_{\pm 0.41}$	**68.34** 66.38$_{\pm 0.69}$	**59.24** 46.55$_{\pm 4.24}$	**68.96** 66.23$_{\pm 1.24}$	61.86 62.02$_{\pm 0.55}$
Story Cloze	**75.33** 72.50$_{\pm 2.53}$	74.12 73.77$_{\pm 0.41}$	**67.47** 65.25$_{\pm 1.52}$	51.36 47.15$_{\pm 4.90}$	**69.92** 66.23$_{\pm 1.24}$	62.33 62.02$_{\pm 0.55}$
SMS Spam	73.88 75.87$_{\pm 1.64}$	72.78 73.77$_{\pm 0.41}$	**67.25** 65.25$_{\pm 1.52}$	**64.69** 46.55$_{\pm 4.24}$	63.70 66.33$_{\pm 1.34}$	**62.13** 61.75$_{\pm 0.26}$
Sentiment 140	**77.56** 73.49$_{\pm 2.33}$	72.78 73.77$_{\pm 0.41}$	66.78 66.38$_{\pm 0.69}$	**51.64** 44.52$_{\pm 2.45}$	66.66 66.00$_{\pm 1.41}$	**62.93** 61.64$_{\pm 0.21}$
TweetQA	**75.22** 72.50$_{\pm 2.53}$	71.52 73.01$_{\pm 1.42}$	66.27 64.91$_{\pm 2.01}$	43.09 44.52$_{\pm 2.45}$	66.76 66.33$_{\pm 1.34}$	61.31 61.33$_{\pm 0.80}$

Table 1: Evaluation of supportive pretraining data to ICL. We obtain supportive pretraining data using the guidance of a *source* task and *evaluate* ICL on all tasks. In the headers, we show the ICL performance of the original LM. We perform perturbative continued pretraining with both our selected supportive data (top number in cells) and an equal number of randomly sampled pretraining data (bottom number in cells). Diagonal cells indicate same-task evaluation and are marked purple. Our performance is bolded when the difference exceeds one standard deviation. On 4 of 6 tasks, the same-task ICL performance gain is observed (diagonal). On 5 of 6 tasks, the corresponding supportive pretraining data improves ICL on at least three tasks (rows).

Zero-shot Eval	Original	+ICL-supportive
SST-2	46.82	46.83
AG News	46.14	44.05
Story Cloze	50.43	51.39
SMS Spam	44.41	43.84
Sentiment 140	55.84	54.90
TweetQA	50.44	50.32

Table 2: Control evaluation. We report the zero-shot prompting performance of the original LM and the perturbed LM after trained on our selected supportive pretraining data. No significant performance gain is observed for most tasks, showing our selected supportive pretraining data is specific to ICL without improving the zero-shot, no-demonstration task performance.

tasks, the token frequency distribution, and the information gain of incorporating long-range context.

3.1 Domain relevance

Xie et al. (2022) and Min et al. (2022) imply that in-context demonstration is useful since it helps locate a particular domain or concept of the test input the LM already learned through the pretraining data. On the other hand, Olsson et al. (2022) imply that in-context demonstration is useful because the decision over the test input may be done through a soft-copy mechanism from the demonstration examples. These lead to two different expectations of the role of supportive pretraining data: (1) Inferred from Xie et al. (2022) and Min et al. (2022), the supportive pretraining data should be from a same domain as the demonstration and test examples, providing direct supporting knowledge to solve the downstream task. (2) Inferred from Olsson et al. (2022), the supportive pretraining data should be beneficial to the soft-copy mechanism, providing meta support for the abstract ability, unconstrained with the concrete data domain.[9] We aim to measure the domain relevance between supportive pretraining data and downstream tasks.

Method To quantify domain relevance, we use MAUVE score (Pillutla et al., 2021) to measure an information divergence between two text distributions. We compute two MAUVE scores, between the target task data and our selected supportive pretraining data, and between the task data and ran-

[9]This view of supportive data will be revisited in §3.3.

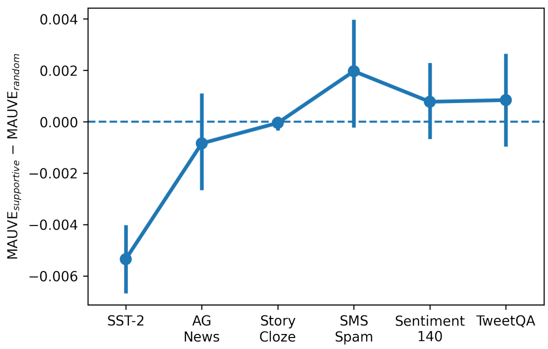

Figure 3: The MAUVE score between the supportive pretraining data and target task data, subtracted by the MAUVE score between random data and target task data. The error bars indicate the 95% confidence interval. No tasks show the supportive data has a significant higher domain relevance compared to random data.

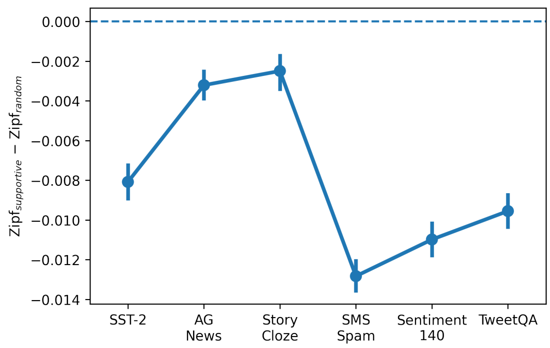

Figure 4: The difference in average Zipf's coefficients of the token frequency distribution of supportive pretraining instances and random examples. The error bars indicate the 95% confidence interval. We find a lower Zipf's coefficient for supportive pretraining data, indicating a flatter frequency distribution, with a relatively higher mass on the rare, long-tail tokens.

dom pretraining data. We then compute and report their difference. A positive MAUVE difference indicates a higher domain relevance of our supportive pretraining data.[10] We use RoBERTa (Liu et al., 2019) as MAUVE's embedding model following He et al. (2022).

Results We show the difference of MAUVE scores in Figure 3. The error bar shows the 95% confidence interval using 32 random seeds. We find that for 5 of the 6 tasks, there is no significant difference between the MAUVE scores of supportive pretraining data and random data. For SST-2, the supportive pretraining data even shows a lower MAUVE score. Therefore, **the supportive pretraining data to ICL do *not* have a higher domain relevance to the task, compared to general pretraining data**. This result aligns with the domain relevance finding in Shin et al. (2022) where dataset-level analyses were performed. This implies the improved ICL behavior of our models may be a meta ability, aided by pretraining data unrelated to the specific domain knowledge for solving the task, but related to a domain-invariant mechanism to learn from a data's context. §3.3 continues this discussion.

3.2 Token frequency distribution

Providing demonstrations to a task input under an ICL setup creates repetitions (e.g., of label tokens), which changes the token frequency distribution of the ICL task data. Therefore, we are interested in whether the supportive pretraining data possess a different token frequency distribution from general pretraining data. Experimented with sequences of image-label pairs, Chan et al. (2022) find that a skewed class distribution (high burstiness) and a large number of rarely occurring classes in training data promote the ICL ability of Transformer models (Vaswani et al., 2017). However, it is unknown whether the findings on the synthetic image-label data can transfer to the natural language pretraining data, a gap we address in this subsection.

Method We fit a Zipfian distribution over each supportive and random pretraining instance that consists of 2048 tokens. The Zipf's coefficient is the negative slope of a linear regression over the tokens' log-rank v.s. log-frequency. A higher Zipf's coeffcient indicates a higher mass on the frequent tokens (i.e., more skewed distribution). A lower Zipf's coefficient indicates a higher mass on the rare, long-tail tokens (i.e., flatter distribution).

Results In Figure 4, we show the difference in average Zipf's coefficients between supportive and random pretraining data, each with a group size of 2000. The error bar shows the 95% confidence interval with 32 random seeds. We find that for all tasks, the Zipf's coefficient of the supportive pretraining data is significantly *lower* than that of the random pretraining data. This indicates a flatter Zipfian distribution with a relatively higher mass over the long-tail tokens. In other words, though the overall burstiness of data is lower, **there is a relatively higher amount of rarely occurring, long-**

[10] Pillutla et al. (2021) also shows higher MAUVE indicates higher generation quality, but we skip that aspect since all of our data are naturally occuring text.

tail tokens in the supportive pretraining data for ICL. Flatter frequency distribution also indicates higher entropy over the tokens, presumably making the supportive pretraining data *challenging* examples to fit by the model, a concept we explore further in the next subsection.

3.3 Information gain from long-range context

In §3.1, we find that the domain relevance of the supportive pretraining data to downstream tasks is not higher than that of random pretraining data. This is comprehendible if we follow the aforementioned perspective of Olsson et al. (2022), hypothesizing that there exists a soft-copy mechanism between the in-context demonstrations and test input. The supportive pretraining data may provide meta support for the abstract soft-copy mechanism rather than task-specific knowledge. We further hypothesize that to facilitate such meta support, the incorporation of long-range context during language modeling in supportive pretraining data should be different from random pretraining data, since the demonstration examples in the ICL setup is a form of long-range context. We propose a novel information gain measure to quantify this feature of incorporating long-range context.

Method Recall that the canonical definition of information gain (IG) is $\mathrm{IG}(T, a) = H(T) - H(T \mid a)$, where T is a target variable, a is an attribute conditioned on by T, and $H(\cdot)$ computes entropy. It measures the decrease of entropy (thus the gain of information) in T if conditioned on a. We adapt the canonical IG to measure the decrease of cross entropy for each token (w_i) in a pretraining dataset when conditioned on a long (l) context over a short (s) context:

$$\mathrm{IG}(l, s) = \mathrm{CE}(w_i \mid \mathrm{ctx}_s) - \mathrm{CE}(w_i \mid \mathrm{ctx}_l)$$

Ideally the length of long or short context should remain constant across different tokens w_i, but it would be a very expensive computation due to a lack of parallelism. We approximate the computation by splitting a full sequence of pretraining tokens (e.g., 2048 tokens) to smaller blocks and calculate cross entropy with the boundary of blocks:

$$\begin{aligned}\mathrm{IG}(l, s) = &- \log p_\theta(w_i \mid w_{i-(i \bmod 2s)\,:\,i}) \\ &+ \log p_\theta(w_i \mid w_{i-(i \bmod 2l)\,:\,i})\end{aligned}$$

With the above definition, the average length of context for all w_i is s and l, respectively. In the experiments below, we keep $s = 128$ for the length of short context and increase the length of long context at $l = \{256, 512, 1024\}$.

We report the difference in the average information gain (across w_i) of incorporating long-range context for a language modeling objective, in supportive pretraining data over random pretraining data. Additionally, we want to use the defined information gain measure as a standalone feature of data, so we use a different LM to compute the cross entropy than the LM on which we perform ICL. Below we report results using OPT-1.3B, while experiments using OPT-350M shows a similar trend.

Results In Figure 5, we see for all of the experimented tasks, there is a significant trend that increasing the length l for the long-range context for supportive pretraining data has a *lower* relative information gain compared to random pretraining data. Though seeming counterintuitive at first glance, this suggests that **the supportive pretraining data are more *challenging* examples in incorporating the long-range context information**.[11] A possible explanation for this is that such challenging examples contain confounding spans that harms the information gain measure. The language model has to learn to decide which part of the long-range context is truly relevant to the prediction of next tokens. This would resemble more and thus helpful to the ICL task scenario where there are multiple demonstrations from different classes.

3.4 Future work

Despite our aforementioned findings, we mainly conduct correlational analyses throughout the work. Despite the potential confounding factors, future work can try converting the correlational findings to causal ones. For example, to actively refine or construct pretraining data to improve existing models' ICL performance, with a metric of token frequency distribution (i.e., find data with a higher mass of long-tail tokens) or context information gain (i.e., find difficult examples in incorporating long-range context). Additionally, we only investigate classification tasks in this work. However, the ORCA-ICL method can be applicable to generation tasks as well in the future, if the ICL loss is defined over a sequence probability of the generation.

[11] Note that a reverse of the statement may not hold necessarily, since an example's long-range context can also be irrelevant by nature and challenging in a useless way.

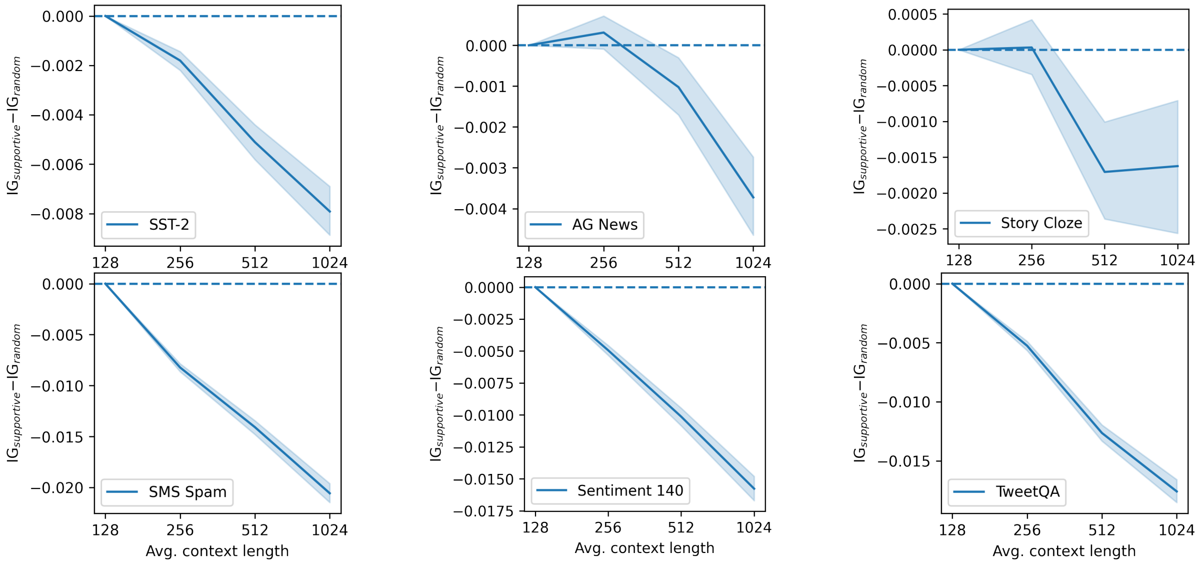

Figure 5: The difference between supportive pretraining instances and random examples in information gain of incorporating long-range context for next-token prediction. We fix the average short context length (s) at 128 tokens and iterate through long context lengths (l) of $\{256, 512, 1024\}$. The shaded area shows the 95% confidence interval. The results show that the long-range context in supportive pretraining data leads to a lower information gain than random pretraining examples. Supportive pretraining data are *challenging* examples in incorporating their long-range context.

4 Related Work

Demonstration examples Min et al. (2022) understand ICL through analyzing which aspects of the demonstration examples contribute or are irrelevant to task performance. They find replacing ground truth demonstration labels with random labels would not hurt task performance, while ICL still benefits from knowing the label space, distribution of inputs, and sequence format specified in demonstration examples.[12] Zhang et al. (2022a) further show on sequence labeling tasks, the length of demonstrations and the relevance of their tokens are important for ICL.

Learning mechanism Xie et al. (2022) explain ICL as implicit Bayesian inference, occurring when language models infer a shared latent concept from demonstration examples at inference time. They show language models exhibit such ICL behavior by constructing synthetic pretraining data with a controlled distribution of concepts. Garg et al. (2022) empirically show that Transformer models can be trained to learn unseen linear functions from in-context demonstration examples. Olsson et al. (2022) present evidence that multi-layer attention-based models form an induction head and perform ICL by a pattern copying behavior from the prefixing context. More recent work like Akyürek et al. (2022), Dai et al. (2022), and von Oswald et al. (2022) explain ICL in Transformer models as a kind of standard learning algorithms over the demonstration examples, such as gradient descent and regression.

Pretraining data Razeghi et al. (2022) find on numerical reasoning tasks, a language model's ICL performance is highly correlated with the term frequency of the input data in the pretraining corpus. Shin et al. (2022) investigate how ICL can be affected when the pretraining dataset varies. They discover that ICL heavily depends on the corpus domain source, but pretraining with a corpus related to a downstream task does not always translate to a competitive ICL performance on the task. Chan et al. (2022) experiment on a synthetic image-label pairs dataset. They show certain distributional properties of the synthetic pretraining data, such as the burstiness of classes and large numbers of rarely occurring classes, promote the emergence of ICL. Our work belongs to this line of work, but offers a first step towards understanding ICL in realistic NLP tasks through analyzing instance-level pretraining data. Additionally, concurrent to our work, Gu et al. (2023) propose a method that groups pre-

[12]Recent work like Wei et al. (2023) and Pan et al. (2023) show the related findings would depend on model scales as well.

training data by their instrinsic tasks, enhancing instead of interpreting existing language models' ICL ability.

5 Conclusion

In-context learning has shown superior performance on a range of NLP tasks, yet it remained unclear *from where* language models acquired this ability. We approach the problem by identifying a small subset of pretraining data that particularly supports language models to do in-context learning on downstream tasks. We analyze common features of the supportive instances in contrast to general pretraining data and find that: (1) The supportive pretraining data do *not* have a higher domain relevance to the downstream tasks. (2) The supportive data contain a relatively larger amount of rare, long-tail tokens. (3) The supportive pretraining data are more *challenging* instances in incorporating long-range context in language modeling. Our findings may be beneficial to future work that refine or construct pretraining data, in order to actively improve existing models' in-context learning performance.

Limitations

It is worth noting that the supportive pretraining data we investigated throughout the work is w.r.t. the *current* LM, such that a perturbative continued pretraining with the supportive data would improve the final LM checkpoint deployed to downstream tasks. It is possible that for some data which we did not determine as supportive, they *had been* supportive w.r.t. early checkpoints of the LM. With more computing resources, future work may investigate the trend of supportive patterns across multiple checkpoints of a LM throughout the pretraining process.

Additionally, another significant limitation of our work is the amount of involved computing resource. The ORCA-ICL method is gradient-based that requires back-propagation. Since we iterate through a large size of pretraining data, the cost of computation is similar to training a language model with a batch size of 1 on the considered pretraining data. On our 4 nodes each consists of 8 Nvidia V100 GPUs, finding the supportive pretraining data for *each* source task in our experiment would take about a week. One mitigating aspect of such computation is that the gradient calculation can be done asynchronously, therefore enabling the use of idle, leftover GPUs scattered across a cluster of nodes. We plan to explore efficient computation of gradient similarity or move from a paradigm of extracting supportive data to generating supportive data in future work.

Acknowledgements

We thank Naman Goyal, Anjali Sridhar, Zeyu Liu, Victoria Lin, Mengzhou Xia, Weijia Shi, Jiacheng Liu, and Tianxing He for helpful discussions. We also thank the anonymous ACL reviewers and all members of TsvetShop for the valuable feedback. This research is supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via the HIATUS Program contract 2022-22072200004. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

References

Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. 2022. What learning algorithm is in-context learning? investigations with linear models. *arXiv preprint arXiv:2211.15661*.

Tiago A Almeida, José María G Hidalgo, and Akebo Yamakami. 2011. Contributions to the study of sms spam filtering: new collection and results. In *Proceedings of the 11th ACM symposium on Document engineering*, pages 259–262.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Stephanie CY Chan, Adam Santoro, Andrew Kyle Lampinen, Jane X Wang, Aaditya K Singh, Pierre Harvey Richemond, James McClelland, and Felix Hill. 2022. Data distributional properties drive emergent in-context learning in transformers. In *Advances in Neural Information Processing Systems*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Damai Dai, Yutao Sun, Li Dong, Yaru Hao, Zhifang Sui, and Furu Wei. 2022. Why can gpt learn in-context? language models secretly perform gradient descent as meta optimizers. *arXiv preprint arXiv:2212.10559*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc. NAACL-HLT*.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. 2022. A survey for in-context learning. *arXiv preprint arXiv:2301.00234*.

Shivam Garg, Dimitris Tsipras, Percy Liang, and Gregory Valiant. 2022. What can transformers learn in-context? a case study of simple function classes. *arXiv preprint arXiv:2208.01066*.

Alec Go, Richa Bhayani, and Lei Huang. 2009. Twitter sentiment classification using distant supervision. *CS224N project report, Stanford*, 1(12):2009.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2023. Pre-training to learn in context.

Kelvin Guu, Albert Webson, Elizabeth-Jane Pavlick, Lucas Dixon, Ian Tenney, and Tolga Bolukbasi. 2023. Simfluence: Modeling the influence of individual training examples by simulating training runs. *ArXiv*, abs/2303.08114.

Xiaochuang Han and Yulia Tsvetkov. 2022. Orca: Interpreting prompted language models via locating supporting data evidence in the ocean of pretraining data. *arXiv preprint arXiv:2205.12600*.

Tianxing He, Jingyu Zhang, Tianle Wang, Sachin Kumar, Kyunghyun Cho, James Glass, and Yulia Tsvetkov. 2022. On the blind spots of model-based evaluation metrics for text generation. *arXiv preprint arXiv:2212.10020*.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. 2022. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Pang Wei Koh and Percy Liang. 2017. Understanding black-box predictions via influence functions. In *Proc. ICML*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? In *EMNLP*.

Nasrin Mostafazadeh, Michael Roth, Annie Louis, Nathanael Chambers, and James Allen. 2017. Lsdsem 2017 shared task: The story cloze test. In *Proceedings of the 2nd Workshop on Linking Models of Lexical, Sentential and Discourse-level Semantics*, pages 46–51.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, et al. 2022. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*.

Myle Ott, Sam Shleifer, Min Xu, Priya Goyal, Quentin Duval, and Vittorio Caggiano. 2021. Fully sharded data parallel: faster ai training with fewer gpus. https://engineering.fb.com/2021/07/15/open-source/fsdp/.

Jane Pan, Tianyu Gao, Howard Chen, and Danqi Chen. 2023. What in-context learning"learns"in-context: Disentangling task recognition and task learning.

Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. 2019. Language models as knowledge bases? In *Proc. EMNLP*.

Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaïd Harchaoui. 2021. Mauve: Measuring the gap between neural text and human text using divergence frontiers. In *Proc. NeurIPS*.

Garima Pruthi, Frederick Liu, Mukund Sundararajan, and Satyen Kale. 2020. Estimating training data influence by tracking gradient descent. In *Proc. NeurIPS*.

Yasaman Razeghi, Robert L Logan IV, Matt Gardner, and Sameer Singh. 2022. Impact of pretraining term frequencies on few-shot reasoning. *ArXiv*, abs/2202.07206.

Timo Schick and Hinrich Schütze. 2021. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proc. EACL*.

Seongjin Shin, Sang-Woo Lee, Hwijeen Ahn, Sungdong Kim, HyoungSeok Kim, Boseop Kim, Kyunghyun Cho, Gichang Lee, Woomyoung Park, Jung-Woo Ha, et al. 2022. On the effect of pretraining corpora on in-context learning by a large-scale language model. *arXiv preprint arXiv:2204.13509*.

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. 2019. Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Johannes von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. 2022. Transformers learn in-context by gradient descent. *arXiv preprint arXiv:2212.07677*.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, et al. 2022. Benchmarking generalization via in-context instructions on 1,600+ language tasks. *arXiv preprint arXiv:2204.07705*.

Jerry W. Wei, Jason Wei, Yi Tay, Dustin Tran, Albert Webson, Yifeng Lu, Xinyun Chen, Hanxiao Liu, Da Huang, Denny Zhou, and Tengyu Ma. 2023. Larger language models do in-context learning differently. *ArXiv*, abs/2303.03846.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2022. An explanation of in-context learning as implicit bayesian inference. In *International Conference on Learning Representations*.

Wenhan Xiong, Jiawei Wu, Hong Wang, Vivek Kulkarni, Mo Yu, Shiyu Chang, Xiaoxiao Guo, and William Yang Wang. 2019. Tweetqa: A social media focused question answering dataset. *arXiv preprint arXiv:1907.06292*.

Hongxin Zhang, Yanzhe Zhang, Ruiyi Zhang, and Diyi Yang. 2022a. Robustness of demonstration-based learning under limited data scenario. *arXiv preprint arXiv:2210.10693*.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022b. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Proc. NeurIPS*.

A Qualitative examples

In Table 3, we show some qualitative examples of the supportive pretraining data to ICL and random pretraining data. Note that these are illustrative examples extracted from long pretraining instances (each instance consists of 2048 tokens), for a better understandability of our findings. A manual examination of such data is difficult, and we thus propose the quantitative analyses described in the main paper.

Supportive pretraining data to ICL
... Samsung's new Odyssey+ headset could fix its muddled VR vision As one of the world's most technologically innovative companies, Samsung should be leading the pack in VR - one of the decade's top transformative technologies. Instead, it has largely let Microsoft and Facebook determine its role in the VR space, leading to its current situation as an also-ran. If I was betting on whether that will change anytime soon, an FCC leak of the company's new Odyssey+ VR headset (discovered by RoadtoVR) would point to "no." Most of the specs are staying the same as its prior, Windows-dependent Odyssey model: Each eye still gets a 3.5-inch screen with 1,440 by 1,600 resolution, combining for a 110-degree field of view, and AMOLED technology will be used to guarantee dark blacks and rich colors. There's one mystery in the new specs, namely a reference to the AMOLED screens now including something called "SFS." ...
Random pretraining data
... Bangladesh authorities and intelligence officials have long been saying that many of the refugees are involved in illicit drug trade, smuggling, robbery and ransom-seeking. Earlier Tuesday, the elite security agency Rapid Action Battalion arrested nine refugees suspected of being involved in various criminal activities. They had firearms, bullets and sharp weapons, Islam said. Local media reported that Tuesday's chaos began after the arrest of the suspects as one group blamed another for helping the security agency in detaining them. Human rights groups that are involved in the camps acknowledge there are criminal elements among the Rohingya refugees. ...

Table 3: Qualitative examples of the supportive pretraining data to ICL in the task of SMS spam detection. We also show an example of random pretraining data for comparison. As our finding on domain relevance suggested, neither of the examples are about SMS spam, so the language model may not learn direct knowledge about the task from supportive pretraining data to ICL. Compared to the random data, the supportive data to ICL has some relatively low-frequency tokens appear multiple times (e.g., VR, Odyssey, AMOLED) and the language model may learn some meta-knowledge about ICL (e.g., copying behaviors from the context) based on them. However, such patterns are sparse, noisy, and hard to analyze through manual inspections. We therefore present the quantitative analyses in the main paper.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Page 9

☑ A2. Did you discuss any potential risks of your work?
Page 9

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Section 1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☒ Did you use or create scientific artifacts?

Left blank.

☐ B1. Did you cite the creators of artifacts you used?
No response.

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
No response.

☐ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
No response.

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
No response.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
No response.

☐ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
No response.

C ☑ Did you run computational experiments?

Page 9

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Page 9

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 2.2

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 2, Section 3

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Section 3

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.