# PROGEN: Progressive Zero-shot Dataset Generation via In-context Feedback

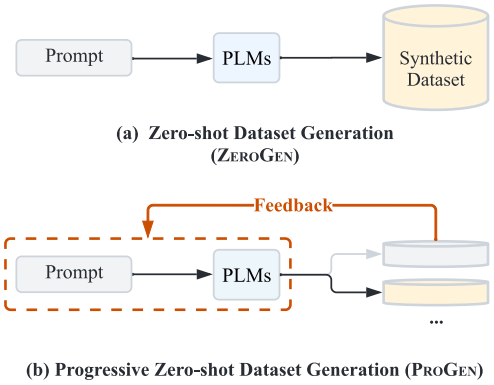
**Jiacheng Ye**[♠◇*], **Jiahui Gao**[♠], **Jiangtao Feng**[◇], **Zhiyong Wu**[◇], **Tao Yu**[♠♡], **Lingpeng Kong**[♠◇]
[◇]Shanghai AI Laboratory [♡]University of Washington
[♠]The University of Hong Kong
{carsonye, sumiler}@connect.hku.hk,
{fengjiangtao,wuzhiyong}@pjlab.org.cn, {tyu,lpk}@cs.hku.hk

## Abstract

Recently, dataset-generation-based zero-shot learning has shown promising results by training a task-specific model with a dataset synthesized from large pre-trained language models (PLMs). The final task-specific model often achieves compatible or even better performance than PLMs under the zero-shot setting, with orders of magnitude fewer parameters. However, synthetic datasets have their drawbacks. They have long been suffering from low-quality issues (e.g., low informativeness and redundancy). This explains why the massive synthetic data does not lead to better performance – a scenario we would expect in the human-labeled data. To improve the quality of dataset synthesis, we propose a progressive zero-shot dataset generation framework, PROGEN, which leverages the feedback from the task-specific model to guide the generation of new training data via in-context examples. Extensive experiments on five text classification datasets demonstrate the effectiveness of the proposed approach. We also show PROGEN achieves on-par or superior performance with only 1% synthetic dataset size compared to baseline methods without in-context feedback.

## 1 Introduction

Dataset generation with pre-trained language models (PLMs) has attracted enormous interest recently due to the superior generative capacity of PLMs. Given task-specific supervision, recent work (Anaby-Tavor et al., 2020; Puri et al., 2020; Kumar et al., 2020; Lee et al., 2021, *inter alia*) manages to fine-tune the PLMs to synthesize high-quality datasets for downstream applications. Nevertheless, obtaining task supervision from human experts can be expensive or even unrealistic. Recent attempts (Schick and Schütze, 2021; Wang et al., 2021; Meng et al., 2022, *inter alia*) turn their eyes to the unsupervised dataset generation.




(a) Zero-shot Dataset Generation (ZEROGEN)

(b) Progressive Zero-shot Dataset Generation (PROGEN)

Figure 1: Comparison of vanilla zero-shot dataset generation (ZEROGEN) and progressive zero-shot dataset generation (PROGEN). In progressive zero-shot dataset generation, we split the whole dataset generation process into multiple phrases. In each phase, the generation is steered by feedback from the previously generated dataset, so as to synthesize a dataset with higher quality.

Among them, ZEROGEN (Ye et al., 2022) proposes to first convert the task descriptions into carefully designed prompts (Petroni et al., 2019; Brown et al., 2020), and then use these prompts to steer the PLMs to synthesize the training data for the final task model. This approach allows highly efficient inference as the final task model only has orders of magnitude fewer parameters compared to PLMs, yet achieves compatible or even better performance than PLMs under the zero-shot setting.

The major drawback of synthetic datasets, however, is they often suffer from low-quality issues (e.g., low informativeness, redundancy). Despite we can generate as much data as computational resource allows, the massive generated data does not automatically translate into better performances, unlike in the human-labeling scenario.

To address this problem, we propose a progressive zero-shot dataset generation framework (Figure 1b), called PROGEN. In a nutshell, PROGEN learns a model for a downstream task by performing two phrases alternatively – using

*Work done while interning at Shanghai AI Lab.



*Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3671–3683
December 7-11, 2022 

PLMs to create labeled examples leveraging the feedback from the current task-specific model, and training a task-specific model given the generated labeled examples. To compute reliable signals as feedback, we employ the influence function (Koh and Liang (2017); IF) to quantify contribution to the loss for each training point. In the context of zero-shot learning where no human-annotated data is assumed, we integrate a noise-resistant objective in the calculation of IF so that it can tackle the noise in the synthetic dataset. To incorporate feedback into PLMs, we sort the training samples based on their quantified influence score, and formulate those most influential ones as in-context examples (Brown et al., 2020) to steer the generation. Overall, PROGEN has the following advantages: 1) the quality estimation phrase requires no human annotations, thus works in a purely zero-shot learning setting; 2) unlike most controllable generation methods that tune or require the access to PLMs (Keskar et al., 2019; Dathathri et al., 2020; Liu et al., 2021, *inter alia*), the in-context feedback phrase does not need to modify parameters in the PLM and incurs minimal disturbance to its generation procedure. Our main contributions are three folds:

- We propose a progressive framework for zero-shot dataset generation to generate higher-quality dataset (§3);
- We propose noise-resistant influence function to estimate the quality of each sample without any human annotations (§3.1), and a learning-free controllable generation method via in-context feedback (§3.2);
- Across multiple text classification datasets, we show our framework obtains better performance over various prompt-based methods, and achieves on-par zero-shot performance with only 1% synthetic dataset size, when compared to methods without in-context feedback (§4).

Our code can be found at https://github.com/HKUNLP/ProGen.

## 2 Background

In this section, we briefly review the baseline approaches of zero-shot dataset generation and how the synthesized dataset can be used for zero-shot learning on downstream tasks.

**Zero-shot Dataset Generation** Take text classification task as an example, vanilla zero-shot dataset generation methods (Meng et al., 2022; Ye et al., 2022) aims to generate a synthetic dataset $\mathcal{D} = \{(\mathbf{x}, y)\}$ with the help of a PLM $\mathcal{P}$. They first sample a class label $y$ from a uniform distribution:

$$y \sim \mathbf{U}(y_1, y_2, \ldots, y_k), \tag{1}$$

where $k$ is the number of classes. They then wrap $y$ up into a label-descriptive prompt $\mathcal{T}(y)$ to steer the generation of $\mathbf{x}$:

$$\mathbf{x} \sim \mathcal{P}(\cdot|\mathcal{T}(y)). \tag{2}$$

Since the parameters of $\mathcal{P}$ is frozen and the generation $\mathbf{x}$ for each $y$ is deterministic, different sampling algorithms (e.g., Top-k sampling (Fan et al., 2018) and nucleus sampling (Holtzman et al., 2020)) can be adopted to increase the diversity of generated dataset. A synthetic dataset is constructed after pairing the generated $\mathbf{x}$ with $y$.

**Dataset-generation-based Zero-shot Learning** The vast linguistic (Jawahar et al., 2019; Goldberg, 2019; Tenney et al., 2019) and factual (Petroni et al., 2019; Jiang et al., 2020b) knowledge encoded in PLMs' parameters is the key towards the success of conventional prompt-based zero-shot learning (PROMPTING) (Brown et al., 2020). However, PROMPTING fails to fully exert the capacity of PLMs and heavily relies on gigantic PLMs during inference. This motivates another line of work (Meng et al., 2022; Ye et al., 2022) to explore a more flexible and efficient way of conducting zero-shot learning based on dataset generation. Given the synthetic dataset generated as above, a task-specific model is trained, allowing any task-specific inductive bias and with an order-of-magnitude smaller number of parameters compared to PLMs. The performance of the final task-specific model is mostly dominated by the quality of the synthetic dataset. A low-quality dataset degrades the final zero-shot performance.

## 3 PROGEN

We now describe our framework for progressive zero-shot dataset generation via in-context feedback (PROGEN), as shown in Figure 2. We follow ZEROGEN (Ye et al., 2022) to build the backbone of our framework. Concretely, we first train a task-specific model (TAM) with partially generated dataset. Then, assuming no access to human

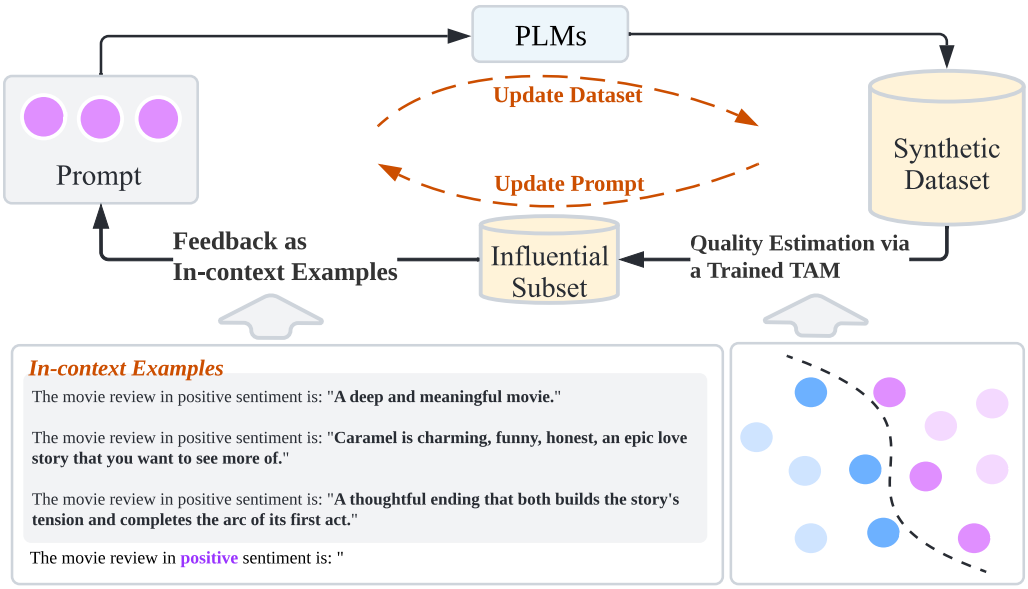



Figure 2: Framework of PROGEN for progressive zero-shot dataset generation. To update the prompt, we first train a task-specific model (TAM) with the synthetic dataset, and then employ the noise-robust influence function to measure the quality of each data point. Finally, the most influential subset is selected, which acts as feedback via in-context learning. The whole framework works with a black-box PLM and requires no human annotations.

annotations, we estimate the influence of each sample via the noise-robust influence function. Finally, with those identified most influential samples, we explore the use of in-context learning to shift the generation distribution towards that of influential samples, so that the system generates more related samples. The whole framework progressively constructs the synthetic dataset and enhances the performance of the final task-specific model.

### 3.1 Annotation-free Quality Estimation

There are many factors in measuring the quality of a dataset, e.g., diversity, annotation correctness, spurious biases (Mishra et al., 2020; Wiegreffe and Marasovic, 2021). However, it is often very subjective, making it unrealistic to calculate them all automatically. Our solution to this is to infer the quality of the individual samples in synthetic datasets using the performance of the final task-specific model trained on the dataset as the surrogate. Concretely, we propose to apply influence function (Koh and Liang, 2017) on the task-specific model to give sample-level influence scores with regard to the loss of validation set. However, a clean validation set, which is crucial for producing reliable influence scores, is inaccessible in the zero-shot learning setting. Thus, we use a synthetic validation set and harness the influence function with a noise-robust objective to handle the potential noise in the synthetic validation set.

Formally, influence function measures the change in the model's loss on the test data-point $z_{\text{test}} = (\mathbf{x}, y)$ if we up-weight the loss of a training data-point $z$ by $\epsilon$:

$$\begin{aligned}\mathcal{I}_{\text{up,loss}}(z, z_{\text{test}}) &\stackrel{\text{def}}{=} \left.\frac{dL\left(z_{\text{test}}, \hat{\theta}_{\epsilon,z}\right)}{d\epsilon}\right|_{\epsilon=0} \\ &= \nabla_\theta L\left(z_{\text{test}}, \hat{\theta}\right)^\top \left.\frac{d\hat{\theta}_{\epsilon,z}}{d\epsilon}\right|_{\epsilon=0} \\ &= -\nabla_\theta L\left(z_{\text{test}}, \hat{\theta}\right)^\top H_{\hat{\theta}}^{-1} \nabla_\theta L(z, \hat{\theta}),\end{aligned} \tag{3}$$

where $\hat{\theta}_{\epsilon,z} \stackrel{\text{def}}{=} \arg\min_{\theta\in\Theta} \frac{1}{n}\sum_{i=1}^{n} L(z_i, \theta) + \epsilon L(z, \theta)$ is the parameter if $z$ were upweighted by some small $\epsilon$ and $H_{\hat{\theta}}$ is the Hessian. Our noise-robust validation-set level influence function is defined as:

$$\mathcal{I}_{\text{up,loss}}(z, \mathcal{D}_{\text{val}}) = -\nabla_\theta L'\left(\mathcal{D}_{\text{val}}, \hat{\theta}\right)^\top H_{\hat{\theta}}^{-1} \nabla_\theta L(z, \hat{\theta}), \tag{4}$$

where $L'$ is a noise-tolerant loss. In this work, we adopt Reverse Cross-Entropy (RCE) loss (Wang et al., 2019), which has the following form:

$$L'\left(\mathcal{D}_{\text{val}}, \hat{\theta}\right) = \sum_{i=1}^{|\mathcal{D}_{\text{val}}|} \ell_{\hat{\theta}}(\hat{y}_i, y_i) = -\sum_{i=1}^{|\mathcal{D}_{\text{val}}|}\sum_{c=1}^{C} \hat{y}_i^c \log(y_i^c), \tag{5}$$

where $\hat{y}_i$ is the predicted class for sample $i$, $C$ is the number of classes.[1] A smaller negative value of

[1] $\log(0)$ is approximated to a constant $A$ in the case of $y_i^c = 0$.

$\mathcal{I}_{\text{up,loss}}$ indicates that upweighting the corresponding training sample will decrease the validation loss more, thus the sample is more valuable. After sorting training samples with $\mathcal{I}_{\text{up,loss}}$ ascendingly, we select top-$M$ (e.g., 50) most valuable samples to form a tiny dataset, which we denote as $\mathcal{D}_{\text{helpful}} = \{(\mathbf{x}, y)\}$.

In practice, we randomly sample a subset of $\mathcal{D}_{\text{train}}$ and adopt the stochastic estimation method described in Koh and Liang (2017) to efficiently compute $\mathcal{I}_{\text{up,loss}}$.

### 3.2 Feedback via In-context Learning

After identifying influence scores for each previously generated sample, we hypothesize that including additional samples similar to those most helpful ones can boost downstream performance. Instead of purely paraphrasing those helpful samples individually, which may hinder the diversity of synthetic dataset, we expect the model to learn from the overall distribution of those helpful samples and generate new samples of similar quality.

Motivated by the striking capability of the in-context Learning (Brown et al., 2020) for PLMs, we propose to use the identified important samples as in-context examples, so that they can shift the generation distribution of PLMs to the ones that are more beneficial to the training of the final task-specific model. Formally, each $\mathbf{x}$ is now generated as follows:

$$\mathbf{x} \sim \mathcal{P}(\cdot|\mathcal{T}(y_1, \mathbf{x}_1), \ldots, \mathcal{T}(y_k, \mathbf{x}_k), \mathcal{T}(y)), \quad (6)$$

where all the in-context examples $\{(\mathbf{x}_i, y_i)\}_{i=1}^{k}$ are randomly selected from $\mathcal{D}_{\text{helpful}}$. Compared with controllable text generation methods that directly modify the parameters in the PLM, we argue that the in-context learning methods incur minimal disturbance to the model's generation procedure.

The overall framework of PROGEN is elaborated in Algorithm 1.

**Algorithm 1** Progressive Zero-shot Dataset Generation

**Require:** a PLM, a TAM, feedback interval $I$, iterations $T$.

$\mathcal{D}_{\text{train}} \leftarrow \varnothing$
$\mathcal{D}_{\text{helpful}} \leftarrow \varnothing$
$\mathcal{D}_{\text{val}} \leftarrow$ Generate a validation set with PLM.
**for** feedback iteration $t = 1, 2 \ldots T$ **do**
  $\mathcal{D}_{\text{new}} \leftarrow$ Generate a dataset of size $I$ with PLM and $\mathcal{D}_{\text{helpful}}$ via Eqn. 6.
  $\mathcal{D}_{\text{train}} \leftarrow \mathcal{D}_{\text{train}} \cup \mathcal{D}_{\text{new}}$.
  TAM $\leftarrow$ Training with $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{val}}$.
  $\mathcal{D}_{\text{helpful}} \leftarrow$ Select most helpful subset from $\mathcal{D}_{\text{train}}$ with TAM and $\mathcal{D}_{\text{val}}$ via Eqn. 4.
**end for**
**Output:** $\mathcal{D}_{\text{train}}$.

## 4 Experiments

### 4.1 Setup

**Datasets** We evaluate our method on five natural language text classification datasets, including IMDb (Maas et al., 2011), SST-2 (Socher et al., 2013), Rotten Tomatoes (Pang and Lee, 2005), Elec (McAuley and Leskovec, 2013) and Yelp (Zhang et al., 2015). Among these datasets, IMDB, SST-2, and Rotten Tomatoes are binary classification benchmarks for movie reviews, Elec and Yelp are binary classification tasks for electronic product reviews and restaurant reviews, respectively. The sizes of the training and test set are 25k/25k, 6.9k/0.8k, 8.5k/1k, 25k/25k, and 560k/38k for IMDb, SST-2, Rotten Tomato, Elec and Yelp, respectively.

**Evaluation Strategy** Following Ye et al. (2022), we evaluate the quality of the synthetic dataset by first training a task-specific model (TAM) with the dataset, and then testing it on a human-annotated dataset (i.e., test set). We also explore other evaluation metrics in § 4.4.

**Baselines** The TAM trained with the synthetic dataset can perform zero-shot inference, hence, we compare PROGEN with other zero-shot learning baselines:

- PROMPTING. The prompt-based zero-shot classification method based via PLMs (Brown et al., 2020).
- PROMPTING*. The calibrated prompting method that reweighs each option according to its priori likelihood (Holtzman et al., 2021).
- ZEROGEN. A recent zero-shot learning work via dataset generation (Ye et al., 2022). They first generate a dataset with a carefully designed instruction, and then train a tiny task-specific model (TAM) to conduct zero-shot inference.

We also provide a non-zero-shot learning baseline SUPERVISED where the same TAM is used but trained on the human-annotated training set.

| TAM | #Param | Setting | IMDb | SST-2 | Rotten Tomato | Elec | Yelp | Avg. |
|---|---|---|---|---|---|---|---|---|
| *#Gold Data* | | | 25k | 6.7k | 8.3k | 25k | 560k | - |
| DistilBERT | 66M | SUPERVISED | 87.24 | 89.68 | 83.67 | 92.63 | 95.42 | 89.73 |
| LSTM | ~7M | | 84.60 | 76.30 | 77.49 | 86.36 | 91.30 | 83.21 |
| - | 1.5B | PROMPTING | 70.50±14.3 | 71.05±26.0 | 68.58±22.2 | 72.76±6.62 | 75.52±10.2 | 71.68±15.9 |
| | | PROMPTING* | 77.31±2.23 | 82.63±8.35 | 78.66±7.23 | 78.03±2.29 | 80.30±6.69 | 79.39±5.36 |
| DistilBERT | 66M | ZEROGEN | 80.41±5.38 | 82.77±6.24 | 78.36±7.68 | 85.35±3.07 | 87.84±2.45 | 82.94±4.96 |
| | | PROGEN | **84.12**±0.26 | **87.20**±1.21 | **82.86**±1.27 | **89.00**±1.16 | **89.39**±0.30 | **86.51**±0.84 |
| LSTM | ~7M | ZEROGEN | 70.18±8.53 | 75.53±10.1 | 72.48±9.36 | 75.84±5.74 | 83.75±2.17 | 75.56±7.19 |
| | | PROGEN | **77.85**±0.84 | **80.96**±1.78 | **77.27**±1.51 | **82.85**±3.17 | **86.03**±1.62 | **80.99**±1.78 |

Table 1: Evaluation results with two different scales of TAM. The scale of synthetic dataset is 100k for both ZEROGEN and PROGEN. We report the average accuracy and corresponding standard deviation across multiple prompts. The detailed results for each prompt are shown in Appendix B.

**Implementation Details** Following Ye et al. (2022), we use GPT2-XL (Radford et al., 2018) and Nucleus Sampling (Holtzman et al., 2020) with $p = 0.9$ for dataset generation. Regarding prompt selection, we adopt a series of prompts for each task. The details of prompt selection are provided in Appendix A. By default, the feedback interval $I$ is set to 1k, and iteration $T$ is set to 100, which ends up with a dataset of size 100k in total. Calculating IF score for all training points is computationally expensive, thus we only sample 10k samples in each iteration. In practice, we find generate data using feedback all the time hinders diversity, thus we only apply feedback half of the time.

We implement an LSTM-based model and a DistilBERT model as TAM to measure the quality of the synthetic dataset. For the LSTM-based model, we use Adam optimizer (Kingma and Ba, 2015), a learning rate of 1e-3, an embedding dim of 100, a hidden size of 300, and a layer number of 1. For DistilBERT, we fine-tune on each dataset with Adam optimizer, with a learning rate of 2e-5, a weight decay of 0.01, and other default hyper-parameters as suggested by HuggingFace Transformers library (Wolf et al., 2019). While using stochastic estimation in the influence function, we randomly sample 10k samples from the whole synthetic dataset, and calculate influence score for those samples over the whole validation set. This operation roughly costs 7 minutes. For all the experiments, we run on a single NVIDIA A100 GPU, and generating 100k examples cost 28h on average for PROGEN.

## 4.2 Main Results

We evaluate the generated datasets by training two different task-specific models and testing their performance on multiple downstream tasks. The results are shown in Table 1. We find that

| | SST-2 | Elec | Yelp |
|---|---|---|---|
| Baseline (ZEROGEN) | 82.77 | 85.35 | 87.84 |
| + syn-random | 86.81 | 87.94 | 87.96 |
| + syn-helpful$_{ce}$ | 86.77 | 87.74 | 89.12 |
| + syn-helpful$_{rce}$(PROGEN) | **87.20** | **89.00** | **89.39** |
| + gold | 90.20 | 91.02 | 91.43 |

Table 2: Evaluation results when harnessing baseline with different types of in-context examples. **syn-random**: random generated samples. **syn-helpful$_{ce}$**: generated samples selected by influence function with cross-entropy loss. **syn-helpful$_{rce}$**: generated samples selected by influence function with reverse cross-entropy loss. **gold**: test set examples. We report the average results across multiple prompts.

PROMPTING suffers from high variance over various prompts in zero-shot learning, and PROMPTING* substantially improves average accuracy and reduces variance across different choices of the prompt through calibration, which is also observed by previous work (Zhao et al., 2021; Holtzman et al., 2021; Min et al., 2022a). Compared with PROMPTING and PROMPTING*, ZEROGEN achieves superior performance by distilling task-related knowledge through dataset generation and handling the downstream tasks with a discriminator rather than a generator. Despite ZEROGEN's success, PROGEN further boosts both average accuracy and variance by improving the quality of the synthesized dataset via a generate-then-feedback framework.

## 4.3 Ablations

One of the main contributions of the proposed progressive zero-shot dataset generation framework is that it incorporates the previously generated dataset as feedback to steer generation. We provide an ablation study over various types of feedback, and summarize the results in Table 2. We find that

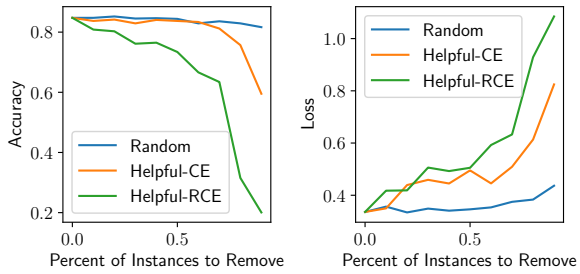



Figure 3: Comparison of various ways to select important examples and the corresponding effects on test set accuracy and loss when removing them. **Helpful-CE**: helpful examples identified by vanilla influence function with cross-entropy loss. **Helpful-RCE**: helpful examples identified by robust influence function with reverse cross-entropy loss.

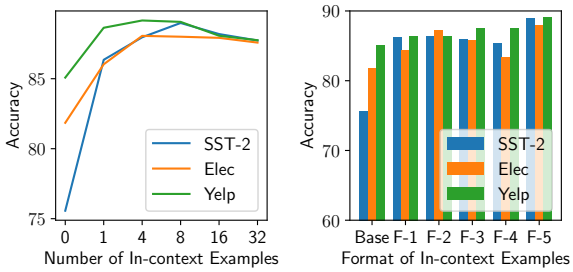



Figure 4: Comparison of different number and format of in-context examples. F-$*$ are different format of in-context examples, see § 4.4 in detail.

providing random synthetic examples as feedback consistently outperforms the baseline method. Our hypothesis is that task-related in-context examples may demonstrate further task-related information than the prompt, and thus benefit the generation process. Selecting examples with vanilla influence function (i.e., CE), only achieves on-par performance with random selection, due to the fact that the signal comes from the noisy validation set is not reliable.[2] In contrast, applying a noise-tolerant objective (i.e., RCE) on the validation set achieves superior performance, which is resistant to the noise in validation set and is able to find more accurate important examples. This proves better task-related signals can further improve the generation quality. Moreover, we find selecting in-context examples from test set examples.[3] obtains the best results, which indicates the model does learn from better in-context examples.

### 4.4 Analysis

**Noise-tolerant influence function provides better estimation in a noisy-validation set scenario.** To see whether using a robust loss function on the validation set contributes to a more accurate estimate, we use a fixed synthetic dataset, remove the estimated important examples, and show how the accuracy and loss of a task-specific model trained with the remained dataset changes. We study three estimation methods, various remove ratios, and evaluate on SST-2 gold test set, as shown in Figure 3. We find filtering random synthetic examples almost does not hurt accuracy and achieves similar accuracy with only 10% examples. Removing helpful examples identified by influence function with cross-entropy increases loss to some extent, but the degree of change is less than using reverse cross-entropy. This shows reverse cross-entropy augmented influence function could offer a more accurate estimate. We also compare results on the test set and artificial noisy test set as validation set in Appendix C, and demonstrate that the two losses are similar when the validation set is clean, but reverse cross-entropy is more effective when the validation set is noisy.

**Format of in-context examples is important.** Given the identified important examples, it's also unknown how to organize these examples as in-context examples. Previous work suggests the order of these examples plays a key role in model performance (Kumar and Talukdar, 2021; Lu et al., 2022), and the performance improves as the number of in-context examples increase (Brown et al., 2020). However, these effects are still under-explored in zero-shot dataset generation.

Suppose we have identified a bunch of positive and negative important examples, and are going to generate with a positive sentiment prompt (e.g., "*The movie review in positive sentiment is: "*"), we study the following formats of in-context examples:

- **Base**: no in-context examples (ZEROGEN).
- **F-1**: positive and negative examples are randomly placed.
- **F-2**: positive examples are placed before negative.
- **F-3**: positive examples are placed after negative.
- **F-4**: only positive examples are placed.

[2] It assumes the validation set objective is the same as the training set.

[3] To prevent trivial solutions (i.e., directly copying in-context examples), we remove the generated texts that are highly overlapped with any given in-context examples.

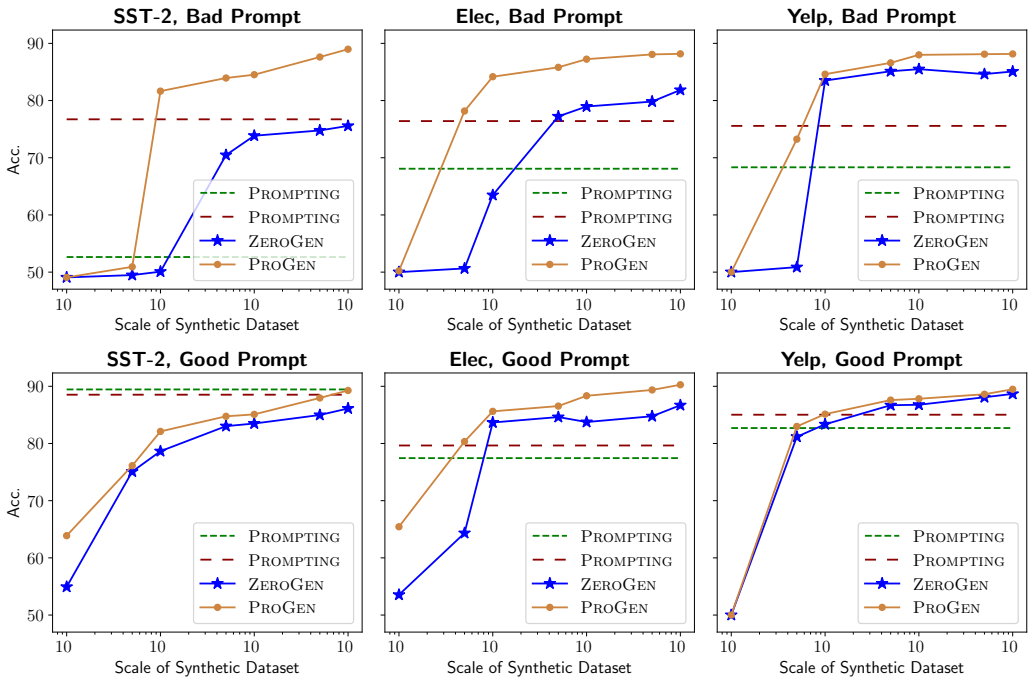



Figure 5: Zero-shot performance with TAM trained under various scale of synthetic dataset.

- **F-5**: only positive examples are placed, but the label information is not expressed (e.g., using prompt "*The movie review is: "<X>"*" for each in-context example, where <X> is the text placeholder to fill in for each in-context example.).

Besides, we also study the number of in-context examples, and the results are shown in Figure 4. We have the following observations. First, a modest number of in-context examples (e.g., no more than 8) consistently improves the performance, however, more in-context examples do not always turn into better performance. We leave the study on larger PLMs (e.g., GPT-3) and PLMs supporting more in-context examples as future work. Second, the performance of different orders of positive and negative examples varies in different tasks, while masking the label information (i.e., F-5) consistently improves performance.

**Prompt selection is less important for PROGEN while scaling dataset is still valuable.** In this part, we visualize the performance change over two representative prompts (i.e., a good prompt $P_2$ and a relatively bad prompt $P_1$) and various scales of dataset, and the comparison is shown in Figure 5.

Overall, we find that PROGEN is more effective for the bad prompt than the good one, and the bad prompt can achieve comparable results with the good one with PROGEN on all the datasets. This indicates the quality of dataset can be iteratively improved with previous lower-quality dataset slice, which shares a similar spirit with Self-Training (Lee et al., 2013) that also learns from its own predictions. Besides, we find PROGEN can achieve similar or superior performance to ZEROGEN with only 1% (100k vs. 1k) size of synthetic dataset. This becomes more meaningful when we only have restricted access to PLMs in real-world applications.

**In-depth analysis of the synthetic dataset.** In previous sections, we measure the quality of the synthetic dataset by training a TAM with that dataset and evaluating on downstream human-annotated data. In this section, we provide other measurements for a more comprehensive understanding of the synthetic dataset. We measure $\mathcal{D}$ from two perspectives, i.e., texts and labels. Regarding texts, we use Self-BELU (Zhu et al., 2018) to measure its own diversity and MAUVE (Pillutla et al., 2021) to measure the similarity between prediction distribution and ground-truth distribution. We measure the correctness of labels with an oracle model trained on human-annotated data as done in Ye et al. (2022). The comparison of different metrics on the Elec dataset is reported in Table 3.

| Granularity | Self-BLEU $f(\mathcal{X})$ | MAUVE $f(\mathcal{X}, \hat{\mathcal{X}})$ | Correctness $f(\mathcal{X}, \mathcal{Y}, \hat{\mathcal{X}}, \hat{\mathcal{Y}})$ | TAM $f(\mathcal{X}, \mathcal{Y}, \hat{\mathcal{X}}, \hat{\mathcal{Y}})$ |
|---|---|---|---|---|
| *Prompt* $P_1$ | | | | |
| ZEROGEN | **19.46** | 79.13 | 62.44 | 81.85 |
| PROGEN | 20.70 | 81.69 | 74.48 | 88.00 |
| PROGEN (Gold) | 27.37 | **92.45** | **89.01** | **90.67** |
| *Prompt* $P_2$ | | | | |
| ZEROGEN | **15.98** | 68.99 | 79.64 | 86.62 |
| PROGEN | 17.66 | 74.64 | 80.86 | 90.27 |
| PROGEN (Gold) | 28.12 | **94.34** | **87.61** | **90.87** |
| *Prompt* $P_3$ | | | | |
| ZEROGEN | **16.08** | 77.61 | 82.99 | 87.58 |
| PROGEN | 19.80 | 80.55 | 84.30 | 88.72 |
| PROGEN (Gold) | 26.16 | **93.75** | **89.59** | **91.52** |

Table 3: Quality of various generated datasets measured by metrics in different granularity. PROGEN (Gold) refers to selecting in-context examples from gold test set. $\mathcal{X}$ and $\hat{\mathcal{X}}$ represent synthetic set and test set, respectively.

Firstly, we find ZEROGEN achieves the highest diversity, and PROGEN degrades diversity. This indicates PLMs can generate texts similar to in-context examples, and the high Self-BLEU score for PROGEN (Gold) is mainly due to the limited number of in-context examples (i.e., 38k for Elec test set vs. 100k for synthetic dataset). Secondly, PROGEN well shifts the generation distribution towards ground-truth distribution, and PROGEN (Gold) achieves significantly higher MAUVE scores. Finally, both PROGEN and PROGEN (Gold) increase label correctness, which is also highly reflected in TAM. Overall, providing feedback improves synthetic datasets in both text distribution and label correctness, but also sightly decreases diversity.

## 5 Related Work

### 5.1 Dataset Generation with PLMs

The accuracy of neural models highly depends on the availability of large-scale human-annotated training data, which, however, can be prohibitively expensive to obtain at scale. Recent advances in generative language models (Radford et al., 2019; Brown et al., 2020) arouse great interests on generating synthetic dataset with PLMs. Some works generate data with a generative model fine-tuned on the public human-annotated dataset (Anaby-Tavor et al., 2020; Puri et al., 2020; Kumar et al., 2020; Lee et al., 2021). Regarding the low quality generations, sample selection (Yang et al., 2020; Liu et al., 2022a) have also been used as postprocessing, which is complementary to our method that improves the dataset quality during generation.

In the context of zero-shot dataset generation, previous approaches adopt prompt-based methods (Jiang et al., 2020a; Shin et al., 2020; Mishra et al., 2021) to generate data without any human-annotations (Schick and Schütze, 2021; Meng et al., 2022; Ye et al., 2022; Gao et al., 2022). The synthetic dataset can be used to train a task-specific model and perform zero-shot inference on downstream tasks. In contrast to our work, all the previous works generate the whole dataset at once, while we consider the quality of previously generated instances and improve the dataset quality during generation.

### 5.2 In-context Learning

Brown et al. (2020) suggest that large PLMs can learn a task by conditioning on a few input-output demonstration pairs as prompt. This paradigm, known as *In-context learning*, is especially attractive as it eliminates the need for updating parameters of the large language model. Subsequent works include better ways of choosing in-context examples (Liu et al., 2022b; Lu et al., 2022; Rubin et al., 2021), learning with an in-context learning objective (Min et al., 2022a; Chen et al., 2022), empirical analysis of why in-context learning works (Min et al., 2022b), theoretical analysis that in-context learning can be formalized as Bayesian inference (Xie et al., 2021), and explorations on other tasks (e.g., semantic parsing (Pasupat et al., 2021), dialogue state tracking (Hu et al., 2022; Xie et al., 2022)). To the best of our knowledge, all previous works study in-context learning in a few-shot learning setting. In contrast, this work focuses on a zero-shot learning setting for dataset generation task and the PLMs' ability to learn from in-context synthetic important examples to produce better dataset.

## 6 Conclusions

This work proposes PROGEN for zero-shot dataset generation, which progressively improves the dataset quality by leveraging feedback from a task-specific model trained on the current dataset. By evaluating zero-shot performance with the trained model, we show PROGEN can generate a much smaller (e.g., 1%) synthetic high-quality dataset that achieves comparable or superior performance to baseline method. We also provide a variety of analyses, including formats of in-context examples, other measurements on synthetic datasets, and the influence of prompt selection.

## Limitations

Our work depends on the PLMs' following abilities: (1) learning from in-context examples; (2) generating relatively high-quality data when using only the manual prompt. This means that if the task can not be well described by a prompt or the PLM is not exposed to enough task-related data in the pre-training stage, the progressive dataset generation process may fail due to the extremely low-quality initial dataset slice and validation set. It can also affect the task-specific model's ability to identify important examples: a noisy validation data point can fool the classifier into trusting mislabeled examples that fall close to it, further degrading the generation quality. In addition, calculating influence function in practice suffers from low-efficiency issues. In practice, we sample a subset from the entire synthesized dataset in each iteration to reduce the computation, which can be sub-optimal in quality estimation accuracy.

## Acknowledgement

We thank the anonymous reviewers whose suggestions helped clarify this work. This work is partially supported by the Shanghai Committee of Science and Technology (Grant No. 21DZ1100100), and the joint research scheme of the National Natural Science Foundation of China (NSFC) and the Research Grants Council (RGC) under grant number N_HKU714/21.

## References

Ateret Anaby-Tavor, Boaz Carmeli, Esther Goldbraich, Amir Kantor, George Kour, Segev Shlomov, Naama Tepper, and Naama Zwerdling. 2020. Do not have enough data? deep learning to the rescue! In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 7383–7390. AAAI Press.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared 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 M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher 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 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Yanda Chen, Ruiqi Zhong, Sheng Zha, George Karypis, and He He. 2022. Meta-learning via language model in-context tuning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 719–730. Association for Computational Linguistics.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2020. Plug and play language models: A simple approach to controlled text generation. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Angela Fan, Mike Lewis, and Yann N. Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 889–898. Association for Computational Linguistics.

Jiahui Gao, Renjie Pi, Yong Lin, Hang Xu, Jiacheng Ye, Zhiyong Wu, Xiaodan Liang, Zhenguo Li, and Lingpeng Kong. 2022. Zerogen$^+$: Self-guided high-quality data generation in efficient zero-shot learning. *CoRR*, abs/2205.12679.

Yoav Goldberg. 2019. Assessing bert's syntactic abilities. *CoRR*, abs/1901.05287.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Ari Holtzman, Peter West, Vered Shwartz, Yejin Choi, and Luke Zettlemoyer. 2021. Surface form competition: Why the highest probability answer isn't always right. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 7038–7051. Association for Computational Linguistics.

Yushi Hu, Chia-Hsuan Lee, Tianbao Xie, Tao Yu, Noah A. Smith, and Mari Ostendorf. 2022. In-context learning for few-shot dialogue state tracking. *CoRR*, abs/2203.08568.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 3651–3657. Association for Computational Linguistics.

Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. 2020a. How can we know when language models know? *CoRR*, abs/2012.00955.

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020b. How can we know what language models know. *Trans. Assoc. Comput. Linguistics*, 8:423–438.

Nitish Shirish Keskar, Bryan McCann, Lav R. Varshney, Caiming Xiong, and Richard Socher. 2019. CTRL: A conditional transformer language model for controllable generation. *CoRR*, abs/1909.05858.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Pang Wei Koh and Percy Liang. 2017. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1885–1894. PMLR.

Sawan Kumar and Partha P. Talukdar. 2021. Reordering examples helps during priming-based few-shot learning. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 4507–4518. Association for Computational Linguistics.

Varun Kumar, Ashutosh Choudhary, and Eunah Cho. 2020. Data augmentation using pre-trained transformer models. *CoRR*, abs/2003.02245.

Dong-Hyun Lee et al. 2013. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 896.

Kenton Lee, Kelvin Guu, Luheng He, Tim Dozat, and Hyung Won Chung. 2021. Neural data augmentation via example extrapolation. *CoRR*, abs/2102.01335.

Alisa Liu, Maarten Sap, Ximing Lu, Swabha Swayamdipta, Chandra Bhagavatula, Noah A. Smith, and Yejin Choi. 2021. Dexperts: Decoding-time controlled text generation with experts and anti-experts. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 6691–6706. Association for Computational Linguistics.

Alisa Liu, Swabha Swayamdipta, Noah A. Smith, and Yejin Choi. 2022a. WANLI: worker and AI collaboration for natural language inference dataset creation. *CoRR*, abs/2201.05955.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022b. What makes good in-context examples for gpt-3? In *Proceedings of Deep Learning Inside Out: The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures, DeeLIO@ACL 2022, Dublin, Ireland and Online, May 27, 2022*, pages 100–114. Association for Computational Linguistics.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 8086–8098. Association for Computational Linguistics.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA*, pages 142–150. The Association for Computer Linguistics.

Julian McAuley and Jure Leskovec. 2013. Hidden factors and hidden topics: Understanding rating dimensions with review text. In *Proceedings of the 7th ACM Conference on Recommender Systems*, RecSys '13, page 165–172, New York, NY, USA. Association for Computing Machinery.

Yu Meng, Jiaxin Huang, Yu Zhang, and Jiawei Han. 2022. Generating training data with language models: Towards zero-shot language understanding. *CoRR*, abs/2202.04538.

Sewon Min, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022a. Noisy channel language model prompting for few-shot text classification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 5316–5330. Association for Computational Linguistics.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022b. Rethinking the role of demonstrations: What makes in-context learning work? *CoRR*, abs/2202.12837.

Swaroop Mishra, Anjana Arunkumar, Bhavdeep Singh Sachdeva, Chris Bryan, and Chitta Baral. 2020. DQI: measuring data quality in NLP. *CoRR*, abs/2005.00816.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, Yejin Choi, and Hannaneh Hajishirzi. 2021. Reframing instructional prompts to gptk's language. *CoRR*, abs/2109.07830.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 115–124, Ann Arbor, Michigan. Association for Computational Linguistics.

Panupong Pasupat, Yuan Zhang, and Kelvin Guu. 2021. Controllable semantic parsing via retrieval augmentation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 7683–7698. Association for Computational Linguistics.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick S. H. Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander H. Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 2463–2473. Association for Computational Linguistics.

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 *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 4816–4828.

Raul Puri, Ryan Spring, Mohammad Shoeybi, Mostofa Patwary, and Bryan Catanzaro. 2020. Training question answering models from synthetic data. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 5811–5826. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2021. Learning to retrieve prompts for in-context learning. *CoRR*, abs/2112.08633.

Timo Schick and Hinrich Schütze. 2021. Generating datasets with pretrained language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 6943–6951. Association for Computational Linguistics.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 4222–4235. Association for Computational Linguistics.

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, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1631–1642. ACL.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R. Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R. Bowman, Dipanjan Das, and Ellie Pavlick. 2019. What do you learn from context? probing for sentence structure in contextualized word representations. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Yisen Wang, Xingjun Ma, Zaiyi Chen, Yuan Luo, Jinfeng Yi, and James Bailey. 2019. Symmetric cross entropy for robust learning with noisy labels. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 322–330.

Zirui Wang, Adams Wei Yu, Orhan Firat, and Yuan Cao. 2021. Towards zero-label language learning. *CoRR*, abs/2109.09193.

Sarah Wiegreffe and Ana Marasovic. 2021. Teach me to explain: A review of datasets for explainable NLP. *CoRR*, abs/2102.12060.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2021. An explanation of in-context learning as implicit bayesian inference. *CoRR*, abs/2111.02080.

Tianbao Xie, Chen Henry Wu, Peng Shi, Ruiqi Zhong, Torsten Scholak, Michihiro Yasunaga, Chien-Sheng Wu, Ming Zhong, Pengcheng Yin, Sida I. Wang, Victor Zhong, Bailin Wang, Chengzu Li, Connor Boyle, Ansong Ni, Ziyu Yao, Dragomir R. Radev, Caiming Xiong, Lingpeng Kong, Rui Zhang, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. 2022. Unifiedskg: Unifying and multi-tasking structured knowledge grounding with text-to-text language models. *CoRR*, abs/2201.05966.

Yiben Yang, Chaitanya Malaviya, Jared Fernandez, Swabha Swayamdipta, Ronan Le Bras, Ji-Ping Wang, Chandra Bhagavatula, Yejin Choi, and Doug Downey. 2020. G-daug: Generative data augmentation for commonsense reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 1008–1025. Association for Computational Linguistics.

Jiacheng Ye, Jiahui Gao, Qintong Li, Hang Xu, Jiangtao Feng, Zhiyong Wu, Tao Yu, and Lingpeng Kong. 2022. Zerogen: Efficient zero-shot learning via dataset generation. *CoRR*, abs/2202.07922.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 12697–12706. PMLR.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A benchmarking platform for text generation models. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*, pages 1097–1100. ACM.

## A Prompt Design

Table 4 summarizes the text prompts used in this work. We choose prompts with the most representative "Control Code" and "Natural Language" styles as discovered by Ye et al. (2022). We also include a two-stage prompt $P_3$ that uses an additional generated text by PLM as condition, e.g., the movie name is generated by prompt "Movie: "".

## B Detailed Results on Each Prompt

The detailed results on each prompt are reported in Table 5 and Table 6.

## C Robust Influence Function on Artificial Noisy Data

To investigate the ability of robust influence function in a noisy-validation set scenario, we create an artificial noisy dataset based on the human-annotated dataset. Specifically, we reverse a portion (e.g., 40%) of ground-truth labels in the training and validation set, and compare the results of using gold validation set and mislabeled validation set when calculating influence function. We show the result comparison in Figure 6, where we remove examples based on the calculated score, retrain the model with the rest of dataset, and evaluate it on a held-out human-annotated set. We find when the validation set is well labeled, both cross entropy-based and reverse cross entropy-based influence function achieve on-par performance. However, a noisy validation set with cross-entropy has a great impact on the quality estimation – removing identified helpful examples only slightly degrades accuracy. In contrast, a noise-resistant objective still provides reliable estimates.

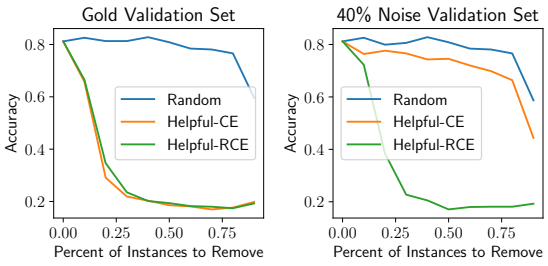



Figure 6: Comparison on artificial noisy dataset.

| Setting | Id | Prompt Details | Label word <Y> |
|---|---|---|---|
| PROMPTING / PROMPTING* | $P_1$ | <Y> <TASK> Review: "<X>" | Positive/Negative |
| | $P_2$ | The <TASK> review in <Y> sentiment is: "<X>" | positive/negative |
| ZEROGEN / PROGEN | $P_1$ | <Y> <TASK> Review: " | Positive/Negative |
| | $P_2$ | The <TASK> review in <Y> sentiment is: " | positive/negative |
| | $P_3$ | The <TASK> review in <Y> sentiment for <TASK> "<C>" is: " | positive/negative |

Table 4: Multiple text prompts for each setting. "<X>" refers to the test input for PROMPTING setting. "<TASK>" represents "movie", "electronic product" and "restaurant" for IMDb/SST-2/Rotten Tomatoes, Elec, Yelp datasets, respectively. "<C>" is a generated text formulated as additional condition to steer generation.

| Setting | Prompt | IMDB | SST-2 | Rotten Tomato | Elec | Yelp | Avg. |
|---|---|---|---|---|---|---|---|
| SUPERVISED | - | 87.24 | 89.68 | 83.67 | 92.63 | 95.42 | 89.73 |
| PROMPTING | $P_1$ | 60.36 | 52.64 | 52.91 | 68.08 | 68.33 | 60.46 |
| | $P_2$ | 80.64 | 89.45 | 84.24 | 77.44 | 82.70 | 82.89 |
| PROMPTING* | $P_1$ | 75.73 | 76.72 | 73.55 | 76.41 | 75.57 | 75.60 |
| | $P_2$ | 78.89 | 88.53 | 83.77 | 79.65 | 85.03 | 83.17 |
| ZEROGEN | $P_1$ | 74.20 | 75.57 | 69.50 | 81.85 | 85.08 | 77.24 |
| | $P_2$ | 83.27 | 86.12 | 82.46 | 86.62 | 88.67 | 85.43 |
| | $P_3$ | 83.76 | 86.61 | 83.11 | 87.58 | 89.76 | 86.16 |
| PROGEN | $P_1$ | 84.22 | 87.16 | 83.02 | 88.00 | 89.06 | 86.29 |
| | $P_2$ | 84.31 | 86.01 | 81.52 | 90.27 | 89.48 | 86.32 |
| | $P_3$ | 83.82 | 88.42 | 84.05 | 88.72 | 89.63 | 86.93 |

Table 5: Results on each prompt with DistilBERT as task-specific model.

| Setting | Prompt | IMDB | SST-2 | Rotten Tomato | Elec | Yelp | Avg. |
|---|---|---|---|---|---|---|---|
| SUPERVISED | - | 84.60 | 76.30 | 77.49 | 86.36 | 91.30 | 83.21 |
| PROMPTING | $P_1$ | 60.36 | 52.64 | 52.91 | 68.08 | 68.33 | 60.46 |
| | $P_2$ | 80.64 | 89.45 | 84.24 | 77.44 | 82.70 | 82.89 |
| PROMPTING* | $P_1$ | 75.73 | 76.72 | 73.55 | 76.41 | 75.57 | 75.60 |
| | $P_2$ | 78.89 | 88.53 | 83.77 | 79.65 | 85.03 | 83.17 |
| ZEROGEN | $P_1$ | 60.33 | 63.88 | 61.73 | 71.16 | 81.29 | 67.68 |
| | $P_2$ | 74.80 | 80.50 | 76.92 | 74.12 | 84.58 | 78.18 |
| | $P_3$ | 75.40 | 82.22 | 78.80 | 82.24 | 85.39 | 80.81 |
| PROGEN | $P_1$ | 77.43 | 81.08 | 76.64 | 79.21 | 84.44 | 79.76 |
| | $P_2$ | 77.30 | 79.13 | 76.17 | 84.38 | 85.97 | 80.59 |
| | $P_3$ | 78.81 | 82.68 | 78.99 | 84.96 | 87.68 | 82.62 |

Table 6: Results on each prompt with LSTM as task-specific model.