ZEROGEN: Efficient Zero-shot Learning via Dataset Generation

Jiacheng Ye[♠◇*†], **Jiahui Gao**[♠†], **Qintong Li**[♠], **Hang Xu**[♣], **Jiangtao Feng**[◇], **Zhiyong Wu**[◇], **Tao Yu**[♠♡], **Lingpeng Kong**[♠◇]

[◇]Shanghai AI Laboratory [♣]Huawei Noah's Ark Lab [♡]University of Washington [♠]The University of Hong Kong

{yejiacheng,fengjiangtao,wuzhiyong}@pjlab.org.cn,
{sumiler,qtli}@connect.hku.hk,xbjxh@live.com,{tyu,lpk}@cs.hku.hk

Abstract

There is a growing interest in dataset generation recently due to the superior generative capacity of large pre-trained language models (PLMs). In this paper, we study a flexible and efficient zero-short learning method, ZEROGEN. Given a zero-shot task, we first generate a dataset from scratch using PLMs in an unsupervised manner. Then, we train a tiny task model (e.g., LSTM) under the supervision of the synthesized dataset. This approach allows highly efficient inference as the final task model only has orders of magnitude fewer parameters comparing to PLMs (e.g., GPT2-XL). Apart from being annotation-free and efficient, we argue that ZEROGEN can also provide useful insights from the perspective of data-free model-agnostic knowledge distillation, and unreferenced text generation evaluation. Experiments and analysis on different NLP tasks, namely, text classification, question answering, and natural language inference, show the effectiveness of ZEROGEN.

1 Introduction

While generating training data with language model is not new to natural language processing (Anaby-Tavor et al., 2020; Puri et al., 2020; Kumar et al., 2020), it has garnered enormous interests recently due to the superior generative capacity of large-scale pre-trained language models (PLMs). Training examples created in such a manner have been found effective in various scenarios via the data augmentation procedure (Lee et al., 2021; Schick and Schütze, 2021; Wang et al., 2021; Meng et al., 2022, *inter alia*).

In this paper, we study an extreme scenario of such an approach, ZEROGEN. Given a downstream task, we first generate its training data from scratch using a powerful PLM, whose generation is steered by carefully designed task-specific prompts. Then, we train a tiny task model (TAM), which has orders of magnitude fewer parameters than PLMs, under the supervision of the synthesized training data. Machine generated text is the only medium that connects the PLMs to the final task models, and no human annotations are required in the entire process. The TAM can be of any choice (e.g., loglinear or neural), allowing efficient inference[1] and deployment. Besides, TAM can be flexibly designed with any task-specific strategies (e.g., inductive bias or loss), which could provide superior performance.

Apart from being annotation-free and efficient, we are also interested in ZEROGEN for the following reasons. First, ZEROGEN can be seen as a variant of knowledge distillation (KD; Hinton et al. (2015)) that provides some exciting new features. Unlike conventional KD, ZEROGEN does not require any human annotations during distillation. Furthermore, ZEROGEN makes no presumption on the architecture choice of student models, thus we can incorporate any task-specific inductive bias into the design of student models conveniently. Second, ZEROGEN can serve as an unreferenced evaluation method for text generation (Guan and Huang, 2020; Pillutla et al., 2021): the downstream tasks' performance is dominated by the quality of the synthesized text, thus can serve as an *indirect* measure of the generation models and algorithms. Third, ZEROGEN sheds new lights on *prompt engineering* (Petroni et al., 2019; Brown et al., 2020) (i.e., the design of the prompts in PLMs). As manual prompts reflect our essential knowledge of specific tasks, an intriguing question here is to what extend we can incorporate human knowledge or instructions in these prompts.

We evaluate ZEROGEN in three NLP tasks which are text classification, question answering, and

*Work done while interning at Shanghai AI Lab.
†Equal Contribution.

[1]Amazon estimates that 90% of production ML infrastructure costs are for inference , rather than training (Jain et al., 2019).

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 11653–11669
December 7-11, 2022

natural language inference, across six datasets. Our key research findings are summarized as follows:

- The zero-shot performance of TAM in ZEROGEN framework significantly surpasses its PLM counterparts (which often serves as the teacher models under the knowledge distillation context), with only ∼0.4% number of parameters (§4.2);
- In some low-resourced settings, TAM trained with synthesized data even outperforms the same model trained with human annotations in a fully supervised manner (§4.3);
- The quality of the generated text by known models and algorithms are well reflected in downstream tasks' performance, and decoding strategies that encourage more diversity also result in greater noise (§4.4);
- Prompt engineering is challenging – the performance of more instructive or natural language style prompts varies in different tasks (§4.5).

In conclusion, we argue that ZEROGEN is a viable and promising approach towards flexible and efficient zero-shot learning in NLP. It also has a great potential as a data-free model-agnostic knowledge distillation and unreferenced text evaluation method. Our code can be found at https://github.com/HKUNLP/ZeroGen.

2 Preliminary: Prompt-based Zero-Shot Learning

We start with preliminary knowledge about prompt-based zero-shot learning framework (named PROMPTING).

Giving a pre-trained language model (PLM) $\mathcal{P}$ and a text classification (TC) task $\mathcal{D} = (\mathcal{X}, \mathcal{Y})$, PROMPTING first instantiates a prompt $\mathcal{T}(\cdot)$ with each input $\mathbf{x}_i \in \mathcal{X}$ and outputs a natural language sequence to be completed by $\mathcal{P}$. For instance, we show an example on sentiment analysis task in Figure 1(a), where $\mathbf{x}_i$ is "*A deep and meaningful film.*" and $\mathcal{T}(\mathbf{x}_i)$ is "*A deep and meaningful film. The sentiment of the movie review is* ". Furthermore, PROMPTING defines a verbalizer $\mathcal{M}(\cdot)$ that maps each label/class y_i to a word/words in $\mathcal{P}$'s vocabulary. For instance, "*positive*" and "*negative*" represents the two classes. In this way, PROMPTING models the probability of class $y_i \in \mathcal{Y}$ for $\mathbf{x}_i$ as:

$$p(y_i|\mathbf{x}_i) = \mathcal{P}\left(\mathcal{M}(y_i)|\mathcal{T}(\mathbf{x}_i)\right). \quad (1)$$

During the whole process, the pre-trained weights of $\mathcal{P}$ are frozen and no training is required.

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 PROMPTING's success. However, PROMPTING fails to fully exert the capacity of PLMs and heavily relies on gigantic PLMs during inference. This motivates us to explore a more flexible and efficient way of conducting zero-shot learning with PLMs.

3 ZEROGEN

In this work, we take the dataset generation method to the extreme and study ZEROGEN, a flexible and efficient zero-shot learning framework via dataset generation. ZEROGEN framework comprises three sequential stages as shown in Figure 1(b):

1. The goal of the first stage is to make use of the generative power of PLMs to synthesize a dataset to solve the downstream task. With carefully designed prompts and a powerful PLM, the generated dataset is believed to incorporate rich task-specific knowledge.
2. Given pseudo dataset synthesized as above, we then train a tiny task model (TAM) to solve the task. TAM can integrate with any task-specific inductive bias and is also order-of-magnitude smaller than PLMs.
3. Finally, we perform efficient inference on target task using the trained model. During the whole process, no human annotations are involved, thus the evaluation setting is purely zero-shot.

Pseudo dataset generation For a single-sentence classification task $\mathcal{D}$, we aim to generate a pseudo dataset $\mathcal{D}^g = (\mathcal{X}^g, \mathcal{Y}^g)$ with the help of a left-to-right PLM $\mathcal{P}$. We first sample a class label y^g from a uniform distribution:

$$y^g \sim \mathbf{U}(y_1, y_2, \ldots, y_k), \quad (2)$$

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

$$\mathbf{x}^g \sim \mathcal{P}(\cdot|\mathcal{T}(y^g)). \quad (3)$$

Since the parameters of $\mathcal{P}$ is frozen and the generation $\mathbf{x}^g$ for each y^g is deterministic, we can

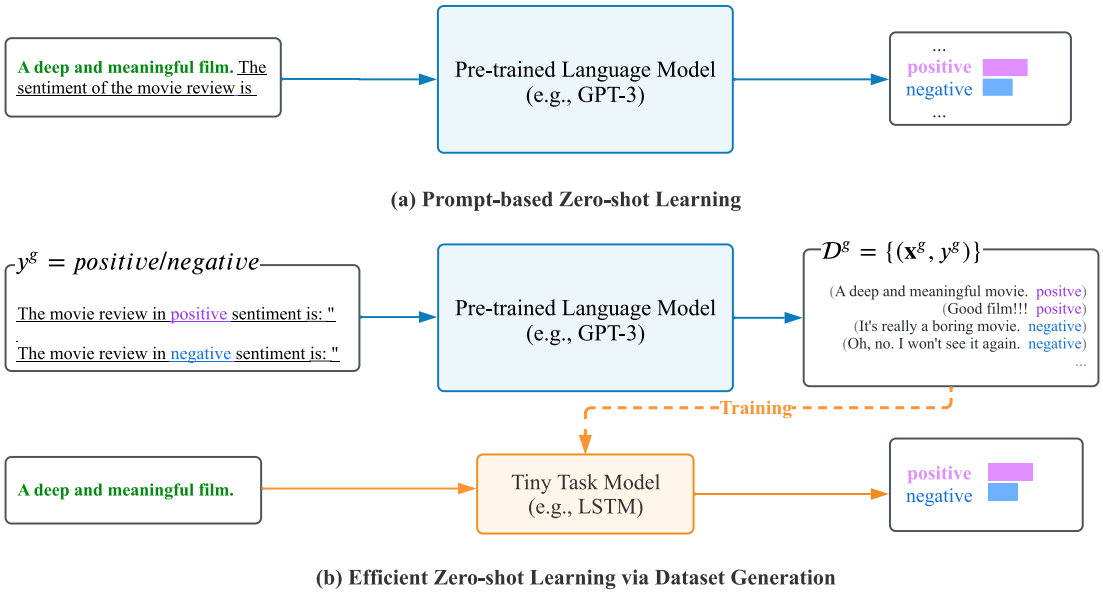

Figure 1: **(a)** Prompt-based zero-shot learning (PROMPTING) framework. The text in **green** is the sentence to be classified. After concatenating the sentence with each prompt, a huge PLM (e.g., GPT-3) is used to calculate LM likelihood score for each class. **(b)** Our ZEROGEN framework. We first generate a training set with PLM in a purely unsupervised manner. After simple filtering operations, we then train a tiny task model (e.g., LSTM) for flexible and efficient inference. The dash line indicates that the training procedure is only performed once before inference.

adopt different sampling algorithms (e.g., Top-k sampling (Fan et al., 2018) and nucleus sampling (Holtzman et al., 2020)) to increase the diversity of generated dataset. We then pair the generated $\mathbf{x}^g$ with y^g to construct a pseudo training data. We show an example about generating a pseudo sentiment classification dataset in Figure 1(b). The prompt $\mathcal{T}(y^g)$ for a positive label y^g is "*The movie review in positive sentiment is: "*". With the sampling strategies, this prompt steers PLMs to generate multiple sentence ending with another quotation mark, e.g., "*A deep and meaningful movie."*" or "*Good film!!!"*".

For sentence-pair classification tasks, we need to generate two sequences that bear certain relationships (e.g., premise and hypothesis in NLI, context and question in QA). We decompose the generation into two steps: (i) We first generate and/or sample a conditional context $\mathbf{c}^g$ (e.g., $\mathbf{c}^g$ represents premise in NLI and context in QA). The context $\mathbf{c}^g$ is then concatenated with a sampled label y_g and transformed into a prompt $\mathcal{T}(\mathbf{c}^g, y_g)$. (ii) Giving the prompt $\mathcal{T}(\mathbf{c}^g, y_g)$, we can now generate the other sentence $\mathbf{x}^g$ (e.g., hypothesis in NLI and question in QA) as in Equation (3). In current implementation, we sample $\mathbf{c}^g$ from an unlabeled corpus. But $\mathbf{c}^g$ can also be generated following procedure of generation for single-sentence classification task. Since there could be no predefined label set for extractive QA task, we use publicly available spaCy[2] toolkit to annotate entities, and then uniformly select an entity as y^g. Finally, the generated sentence-pair and label can form the pseudo dataset $\mathcal{D}^g = (\mathcal{C}^g, \mathcal{X}^g, \mathcal{Y}^g)$. We elaborate details on prompts chosen for each task in Section 4.5.

Pseudo-supervised training With the pseudo dataset $\mathcal{D}^g$, we train a tiny task model TAM to conduct the given task. This procedure is highly flexible, meaning that we can use any model architecture, loss function, and training strategy. In this work, we primarily focus on the overall framework, thus we leave the tuning of these components for future work. Under the zero-shot learning setting, it should be noted that we have no access to the standard validation set. Therefore, we use a portion (e.g., 10%) of the pseudo dataset as the validation set for model selection.

Zero-shot evaluation Finally, we conduct inference on the trained TAM model. As TAM is order-of-magnitude smaller than PLM, it is able to perform extremely efficient inference.

[2]https://spacy.io/

PLM	TAM	#Param	Setting	IMDb	SST-2	SQuAD	AdversarialQA	QNLI	RTE
#Gold Data				25k	6.7k	87k	30k	105k	2.5k
-	DistilBERT	66M	SUPERVISED	87.24	89.68	76.28/84.67	18.6/29.85	88.05	58.12
	LSTM	~7M		84.60	76.30	41.86/57.22	5.37/11.86	69.00	54.87
GPT2	-	117M	PROMPTING	51.52	52.52	0.80/4.93	0.37/2.58	50.60	**52.70**
	DistilBERT	66M	ZEROGEN	**73.24**	**80.39**	**16.44/21.83**	**5.20/8.26**	**55.32**	50.54
	LSTM	~7M		69.60	70.40	4.94/8.53	1.00/3.83	51.03	49.10
GPT2-Large	-	762M	PROMPTING	80.20	**87.84**	3.53/10.78	1.47/5.16	55.10	54.51
	DistilBERT	66M	ZEROGEN	**83.56**	85.44	**23.87/29.82**	**5.93/9.63**	**69.32**	**58.48***
	LSTM	~7M		78.20	75.10	8.01/12.77	2.33/5.24	51.27	56.68*
GPT2-XL	-	1.5B	PROMPTING	80.64	**89.22**	4.61/13.32	2.13/6.30	60.60	57.04
	DistilBERT	66M	ZEROGEN	**84.28**	87.27	**25.50/31.53**	**6.33/9.96**	**71.19**	**59.93***
	LSTM	~7M		79.80	78.40*	12.35/18.66	3.23/6.34	52.26	58.85*

Table 1: Evaluation results for ZEROGEN at three different scales of PLM and two different scales of TAM. The ZEROGEN results that outperform PROMPTING using the same PLM are in grey, and the best result for each task using the same PLM is **bolded**. * indicates that the result of TAM under ZEROGEN framework outperforms the same TAM under SUPERVISED framework. We report the average number of parameters (i.e., 7M) for LSTM-based models among different tasks. The scale of the synthetic dataset is 200k for each task. Results on larger PLMs are reported in §4.6.

4 Experiments

4.1 Setup

We perform experiments across three different tasks including six different NLP datasets. The detailed experimental setup (i.e., Implementation Details) are described in Appendix A.

Datasets We consider two Text Classification datasets (i.e., SST-2 (Socher et al., 2013) and IMDb (Maas et al., 2011)), two Natural Language Inference datasets (i.e., QNLI (Rajpurkar et al., 2016) and RTE (Dagan et al., 2005; Haim et al., 2006; Giampiccolo et al., 2007; Bentivogli et al., 2009)), and two Question Answering datasets (i.e., SQuAD1.1 (Rajpurkar et al., 2016) and AdversarialQA (Bartolo et al., 2020)). The number of training examples for SST-2 and RTE is 6.9k and 2.5k, which can be considered as low resource compared with IMDb (25k), QNLI (105k), SQuAD (87k) and AdversarialQA (30k). We adopt Exact-Match (EM) and F_1 as the metrics for question answering tasks and Accuracy for other tasks.

Baselines We compare ZEROGEN framework with two baselines: **(1)** PROMPTING. The prompt-based zero-shot learning framework via PLMs. We use GPT2 (117M), GPT2-large (762M), and GPT2-XL (1.5B) (Radford et al., 2019) via the HuggingFace Transformers library (Wolf et al., 2019). **(2)** SUPERVISED. The TAMs are trained on standard dataset (i.e., human annotations). Regarding model architecture of TAMs, we use two types of model for each task: a LSTM-based model (i.e., BiLSTM (Hochreiter and Schmidhuber, 1997) for TC and NLI tasks, and BiDAF (Seo et al., 2017) for QA task), and a tiny pre-trained model (i.e., DistilBERT (Sanh et al., 2019)).

Evaluation Strategy Due to restricted test set access for some datasets (i.e., SQuAD1.1 and SST-2), we held out a small subset (i.e., 10%) of the training set for validation for model trainined in SUPERVISED setting, and report results on the validation set. For models trained with synthetic dataset in ZEROGEN framework, we also use a portion (i.e., 10%) as the validation set, without accessing to original validation set. For PROMPTING, we directly evaluate on the original validation set.

4.2 ZEROGEN vs. PROMPTING

Table 1 compares ZEROGEN with PROMPTING framework. We observe that ZEROGEN significantly outperforms PROMPTING on most datasets we evaluated, and this superiority is consistent across different PLM generators and TAMs. In particular, when using DistilBERT as TAM, we find that among 18 (3 generators $\times$ 6 tasks) head-to-head comparison with PROMPTING, ZEROGEN achieves better performance in 15 cases[3]. The reasons for the superior performance are mainly two-folds: 1) compared with the general purpose generation model, task-specific classification model may encourage a more deterministic de-

[3]Note that with careful prompt design and selection, our PROMPTING baseline achieves an accuracy of 89.22% with GPT2-XL on SST-2 dataset, substantially higher than the previous best results (i.e., 87.4% (Holtzman et al., 2021))

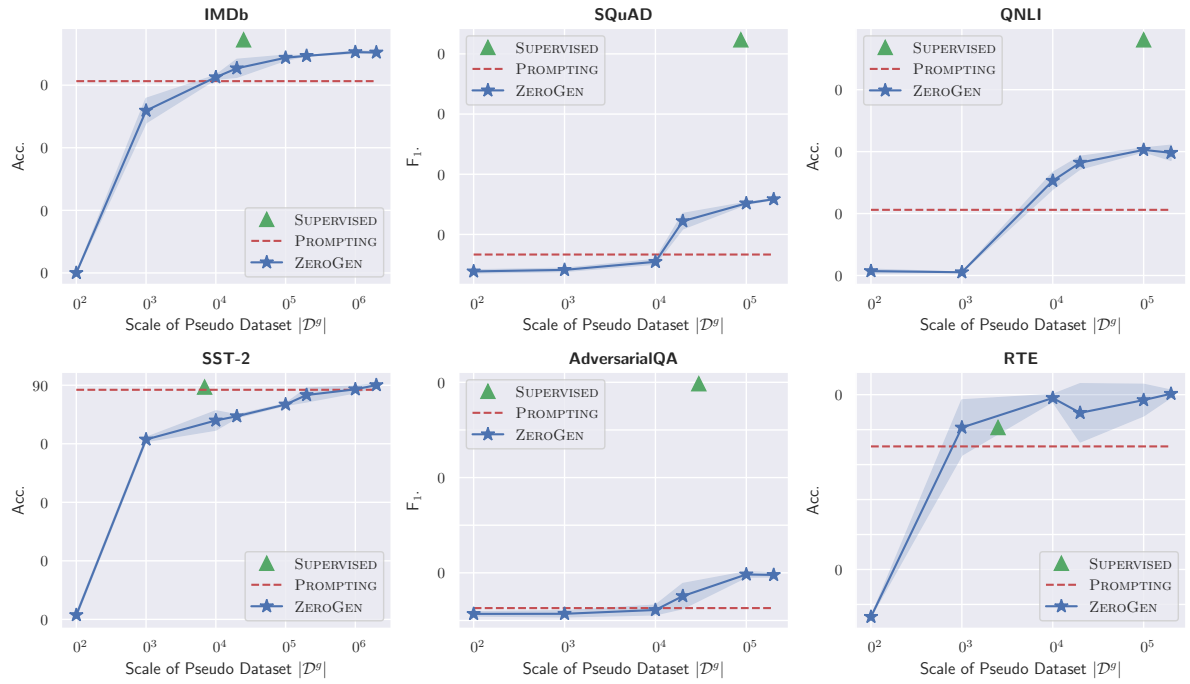

Figure 2: Results for comparing various scales of synthetic datasets on different tasks. We use GPT2-XL as PLM and DistilBERT as TAM. Dots with star marker and error bars are the average performance and the standard deviation over 3 runs, respectively.

cision boundary, which shares the same spirits with entropy minimization (Grandvalet and Bengio, 2006) or self-training (Lee et al., 2013), and 2) classification tasks benefit from the inductive bias in the architecture. For example, method that predicts the start and end positions greatly narrows down the searching space for extractive question answering tasks, in comparison with free-generation on the vocabulary space.

Besides the superior effectiveness in zero-shot learning, it's also worth noting that ZEROGEN is also quite efficient. ZEROGEN can achieve comparable (LSTM) and even better (DistilBERT) performance than PROMPTING, using more than 200 times and 20 times fewer parameters, respectively. Nowadays, with increasingly larger pre-trained language models (e.g., 175B GPT-3 (Brown et al., 2020), 1571B Switch-C (Fedus et al., 2021)), the advantage of ZEROGEN becomes even more pronounced. The gigantic PLMs can improve the quality of synthesized dataset and lead to better zero-shot performance. Meanwhile, the TAM can remain light-weighted for efficient inference and serving.

Furthermore, when scaling up PLMs, we observe continuous performance boost for both PROMPTING and ZEROGEN. This indicates that larger-scale PLMs might have been trained to store more knowledge that is useful for generating accurate dataset for a task.

4.3 ZEROGEN vs. SUPERVISED

It's commonly accepted that the zero-shot performance of a NLP model can lag way behind its fully-supervised performance (trained on human annotations). However, we find that ZEROGEN even outperform its SUPERVISED counterpart on SST-2 and RTE datasets (highlighted with * in the Table 1). Our conjecture is the size of the datasets are the key factor. ZEROGEN automatically generates much more data as supervision during training (i.e., 200k synthesized samples vs. 6.9k/2.5k human annotations). These results are encouraging because they suggest that: (i) ZEROGEN is quite effective in low-resource scenario; (ii) it's possible to synthesize training samples to approximate human-annotations in a fully unsupervised manner.

We further investigate if we can trade data volume in exchange for zero-shot performance in ZEROGEN. Our results are shown in Figure 2.

Overall, we find that the final performance improves continuously as the amount of data grows, despite diminishing returns. We find that generating 10k of training samples leads to better

TAM	Strategy	IMDb	SQuAD	QNLI
DistilBERT	Greedy	74.40	24.19/31.16	63.59
	Top k=5	79.50	**25.68/32.32**	**71.83**
	Top k=40	**83.70**	24.70/31.21	70.40
	Top k=80	**83.70**	24.14/30.89	69.83
	Nucleus p=0.9	83.40	23.93/30.62	70.55
LSTM	Greedy	54.70	10.76/16.24	53.14
	Top k=5	57.30	**13.22/19.45**	**54.68**
	Top k=40	**72.24**	10.11/15.56	51.24
	Top k=80	71.10	9.63/14.70	51.38
	Nucleus p=0.9	71.80	8.83/13.89	51.16

Table 2: Results for comparing different decoding methods with selected parameters of each method. The best result under different decoding methods is **bolded**.

performance than PROMPTING method on most datasets. In addition, by increasing data size, we find that ZEROGEN even outperforms the SUPERVISED baseline on SST-2 and RTE. But still, on some datasets examined (e.g., SQuAD, QNLI), there remain a performance gap between ZEROGEN and SUPERVISED.

4.4 ZEROGEN as Text Generation Evaluator

The quality of the synthesized text is the key to the performance of the downstream tasks. ZEROGEN can thus be seen as an indirect measure of the generation models and algorithms. It is a commonly accepted belief that the quality of generated text should be in an ascending order in GPT-2, GPT2-Large, and GPT2-XL, due to the growing in the parameter size. We find this trend is well aligned in the downstream application performance (Table 1).

Besides the model, another important aspect in text generation is its decoding algorithm, where the goal is to achieve better diversity without the text quality (e.g., fluency, coherence, and correctness). We show that how the trade-off between *diversity* and *correctness* is reflected in the framework of ZEROGEN.

Overall Results Sampling strategies (e.g., top-k sampling and nucleus sampling) are known to be able to generate text with a higher degree of diversity than other decoding strategies (e.g., greedy search) (Fan et al., 2018; Holtzman et al., 2020). Empirical results in Table 2 demonstrate that a more diverse decoding strategy does not always ensure better performance on downstream tasks. For example, the results of the nucleus sampling strategy, which is considered to generate the most diverse data, achieves a performance nearly 6% and 3% lower than the best decoding strategy on both the SQuAD and QNLI datasets, respectively, while greedy decoding strategy could obtain better results than some sampling strategy (e.g., top-k=40, top-k=80 and nucleus sampling). In contrast, all sampling strategies are superior to the greedy decoding strategy on the IMDb dataset. Regarding the inconsistent better downstream performance of more diverse decoding strategies, we hypothesize that diversity may come at a price, such as generating samples not pertain to the class described in the prompt. Therefore, we assess the quality of a dataset from two perspectives: *Diversity* and *Correctness* for quantitative analysis of different datasets.

DIVERSITY			
Strategy	**IMDb**	**SQuAD**	**QNLI**
Oracle	0.30	0.14	0.14
Greedy	0.92	0.55	0.54
Top-k=5	0.59	0.29	0.28
Top-k=40	**0.25**	0.17	**0.14**
Top-k=80	0.20	0.17	0.12
Nucleus p=0.9	0.15	**0.16**	0.11
CORRECTNESS			
Strategy	**IMDb**	**SQuAD**	**QNLI**
Oracle	92.42	95.37	92.96
Greedy	99.67	**31.07**	**83.18**
Top-k=5	**94.46**	18.74	74.03
Top-k=40	84.91	14.57	64.31
Top-k=80	84.11	13.74	62.94
Nucleus p=0.9	82.53	13.30	63.37

Table 3: Diversity and Correctness evaluation of generated datasets under different decoding strategies. Oracle refers to the standard dataset with human annotations. Diversity is measured by Self-BLEU4, while Correctness is measured by a well-trained RoBERTa-Large model with standard dataset.

Diversity We follow previous work (Holtzman et al., 2020) and compute Self-BLEU (Zhu et al., 2018) as a metric of diversity. Self-BLEU is calculated by computing the BLEU score of each generated text using all other generations in the evaluation set as references[4]. A lower Self-BLEU score implies higher diversity. We report 4-gram based Self-BLEU in the first part of Table 3. We find that decoding strategies such as top-k and nucleus sampling lead to more diverse generations. This finding is consistent with previous works (Li et al., 2016; Vijayakumar et al., 2016; Welleck et al., 2020; Holtzman et al., 2020).

[4]Specifically, we randomly sample 1000 generations, each of which is compared with all 999 other generations as references.

Correctness Different from the vanilla generation scenario that ends with the generated text, we use the generated text as training dataset for another small model. Therefore, ZEROGEN requires a more emphasis on the correctness of generated text, i.e., whether the generated text pertain to the corresponding class described in the prompt. To access the correctness of a synthetic dataset, we first train a RoBERTa-Large (Liu et al., 2019) model on the standard training dataset, which is then used as a validator to evaluate the synthetic dataset. In summary, we find a tradeoff between diversity and correctness, i.e., greater diversity leads to lower correctness. We notice even deteriorated outcomes by increasing k, while greedy search achieves the highest performance in terms of correctness. These results reflect those of Massarelli et al. (2020) who also found a tradeoff between factuality and diversity, i.e., while decoding strategies such as top-k and nucleus sampling lead to less repetitive generations, they also produce less verifiable text. Besides, among different tasks, we find the correctness on oracle datasets are similar (i.e., all larger than 90%), while that varies substantially on synthetic datasets (i.e., up-to 94.46% on IMDb and merely 31.07% on SQuAD). Compared with generating datasets for single text classification tasks (e.g., IMDb), where the PLM only needs to consider a single condition (i.e., label), generating for text-pair tasks requires PLMs to consider multiple conditions synchronously (e.g., answer and context when generating question), which makes it more difficult to control the correctness of the generated sample. This possibly explains the observed variance among tasks.

Human Evaluation We report the human evaluation results in Table 4. The quality of generated data is measured by the *correctness* and *naturalness* metrics. The *correctness* measures whether the label is correct and the content is relevant to the task topic (e.g. movie review for IMDb). The *naturalness* measures whether the generated text is fluent and similar to human-generated text. We invite 4 experts to participate in the evaluation and each participant is randomly assigned 25 generated samples (100 samples in total) for each decoding strategy. Table 4 report the mean scores. The results show that greedy search achieves the highest performance in terms of correctness, which is consistent with the automatic evaluation using Roberta-Large. However, in terms of naturalness/fluency, the greedy search performs the worst. The top-k and nucleus decoding strategies can generate a more fluent context by reducing repetitive generation.

Method	Correctness	Naturalness	ZEROGEN
Oracle	0.92	4.46	87.2
Greedy	0.91	3.55	74.4
Top k=40	0.72	3.75	83.7
Nucleus p=0.9	0.81	3.89	83.4

Table 4: Human evaluation on IMDb gold and synthetic dataset using different decoding strategies. We also show the TAM performance to show the ability of ZEROGEN as an evaluator.

4.5 Prompt Engineering in ZEROGEN

The design of prompts can have huge impact on PROMPTING, as pointed by many previous works (Mishra et al., 2021a; Wei et al., 2022). In this section, we investigate how prompt design instructs text generation and affects ZEROGEN's performance. We examine three commonly used prompt types: (1) *Control code* (Keskar et al., 2019), (2) *Control code with task description*, (3) *Natural language style*. For SST-2 and IMDb, example prompts and corresponding results can be found in Table 5 (check Appendix A for other tasks).

From Table 5, we first observe that *natural language* prompts are favored by both ZEROGEN and PROMPTING, rather than prompts contain control code. We hypothesize the reason being that during the pre-training process, the majority of text data fed to the PLMs are natural language sentences, and therefore the PLMs do not contain enough knowledge in control code. Moreover, we observe that ZEROGEN is more robust towards different prompts than PROMPTING: for PROMPTING, a minor change from P_4 to P_5 will lead to a huge drop in accuracy (16.2% drop in IMDb); for ZEROGEN, applying the same prompt revision, the decrement decreases to 9.4%. Compared with PROMPTING which use prompt to directly instruct label words, ZEROGEN use synthesized data as medium to connect PLM and TAM, thus mitigating the sharp change brought by prompts.

To further explore the potential of prompting, we investigate the two-stage conditional prompt inspired by (Schick and Schütze, 2021). In the running example, based on the task characteristic(to generate a movie review), we first generate

Setting	Id	Prompt	Label word <Y>	Prompt Type	IMDb	SST-2
PROMPTING	P_1	"<X>" It was <Y>	great/terrible	-	51.32	51.83
	P_2	<Y> Movie Review: "<X>"	Positive/Negative	Control code	60.36	52.75
	P_3	Task: Write a review for a <Y> movie. Review: "<X>"	good/bad	Control code with task description	54.20	53.50
	P_4	The movie review in <Y> sentiment is "<X>"	positive/negative	Natural language style	80.64	89.22
	P_5	The <Y> movie review is "<X>"			67.60	72.36
ZEROGEN	$P_1^{'}$	It was a review for a <Y> movie: "	great/terrible	-	<u>79.36</u>	<u>75.00</u>
	P_2	<Y> Movie Review: "	Positive/Negative	Control code	<u>60.88</u>	<u>67.43</u>
	P_3	Task: Write a review for a <Y> movie. Review: "	good/bad	Control code with task specification	83.40	78.90
	P_4	The movie review in <Y> sentiment is: "			81.84	86.24
	$P_4^{'}$	The movie review in <Y> sentiment for movie "<C>" is: "	positive/negative	Natural language style	83.40	86.35
	P_5	The <Y> movie review is "			77.44	77.06

Table 5: Results for different prompts on IMDb and SST-2 dev sets. We use GPT2-XL as PLM and DistilBERT as TAM. <X> and <C> represents the input sentence and generated movie name respectively. $P^{'}$ represents minor revised version of P for text generation. For ZEROGEN, results are reported using 100k training samples. Scores in <u>underline</u> are trained on 10k generated samples, since the prompt ($P_1^{'}$ & P_2) is too weak and cannot generate 100k distinct samples.

PLMs	Setting	IMDB	SQuAD	QNLI
GPT2-XL	PROMPTING	80.64	4.61/13.32	60.60
	ZEROGEN-LSTM	79.80	12.35/18.66	51.53
	ZEROGEN-DistilBERT	**84.28**	**25.50/31.53**	**71.19**
OPT	PROMPTING	63.18	23.35/39.32	**54.51**
	ZEROGEN-LSTM	73.08	21.46/30.06	50.76
	ZEROGEN-DistilBERT	**79.99**	**33.27/44.91**	52.97

Table 6: Comparison of GPT2-XL (1.5B) and OPT (175B) under the same prompt and decoding strategy.

movie name using prompt *[Movie: "]* and then prompt sentence using $P_4^{'}$. We can find that with the control of movie name, the generated training corpus is more diverse than using P_4. With the desirable correctness (see Table 3), the higher diversity leads to a higher accuracy (from 81.84 to 83.40 in IMDb).

The most suitable prompting type in Question Answering and Natural Language Inference tasks has some differences with Text Classification due to different task characteristics. For details, please refer to the Appendix B.

4.6 ZEROGEN via Larger PLM Generator

We further investigates the performance of ZEROGEN on a larger PLM (i.e., OPT (Zhang et al., 2022) with 175B parameters). We find both PROMPTING and ZEROGEN benefit from the larger PLM on hard tasks (i.e., SQuAD). But on relatively simpler text classification tasks, the results degrades. This demonstrates that prompt selection is still important for larger models, and the prompt that suits for one model may not suit for another.

5 Related Work

5.1 Prompt-based Zero-shot Learning

With manual crafted natural language prompt, large-scale PLMs have shown impressive zero-shot abilities in a wide array of NLP tasks(Radford et al., 2019; Brown et al., 2020). However, current prompt-based zero-shot learning can be unstable: the choice of prompt contributes a lot to the final performance. This motivates researchers to investigate better ways to automatically search and/or manually construct a proper prompt (Jiang et al., 2020a; Shin et al., 2020; Reynolds and McDonell, 2021; Mishra et al., 2021b). To improve the zero-shot generalization across different prompts, another line of work uses a multitask training mixture made up of a large set of different tasks specified in natural language prompts. This induces a model to better generalize to unseen tasks, as well as being more robust to the wording choices of the prompts. (Khashabi et al., 2020; Zhong et al., 2021; Mishra et al., 2021c; Wei et al., 2021; Sanh et al., 2021; Xu et al., 2022). In comparison, we advocate and analyse a new paradigm for prompt-based zero-shot learning via dataset generation, which is complementary to current prompt searching and multi-task pre-training methods.

5.2 Dataset Generation with PLMs

Our work also relates to research in generating data with PLMs, which aims to generate a pseudo dataset to enhance model performance. Early efforts achieve this goal with fine-tuned generative

models (Anaby-Tavor et al., 2020; Puri et al., 2020; Kumar et al., 2020; Lee et al., 2021). They first fine-tune the generative models using human annotations, the generated data samples are then combined with human annotations to train the models in a semi-supervised fashion. Supervised data generation methods are also studied for building auxiliary tasks (Vu et al., 2021) and dataset creation based on human and machine collaboration (Liu et al., 2022). To reduce the human efforts on data annotation, another line of works explore data generation methods without the need for human annotations. He et al. (2021) uses unsupervised-trained unconditional generative models to synthesize unlabeled data for semi-supervised learning. Without any model training, Wang et al. (2021) propose to directly use unlabeled in-domain examples as prompts to synthesize high-quality training data. Schick and Schütze (2021) explore dataset generation method from scratch for semantic textual similarity task. One concurrent work (Meng et al., 2022) studies dataset generation for text classification and natural language inference tasks. In comparison, we take the dataset generation framework to the extreme, i.e., consider extremely tiny edge models (e.g., LSTM), explore boarder NLP tasks including question answering, and conduct extensive analysis such as decoding strategies and quality evaluation.

6 Conclusion and Future Directions

In this paper, we study an extreme instance of dataset generation via PLMs for zero-shot learning. Without any human annotations, we show that an small LSTM can surpass the zero-shot performance of its PLM counterparts (e.g., GPT2-XL), and even outperform the same model trained with human annotations. Despite the demonstrated effectiveness, we discuss several issues we observed when developing ZEROGEN and reveal a substantial room of improvement in future research.

Despite positive results on TC tasks, we find the stability regarding prompt choice of ZEROGEN is still far from satisfactory on NLI tasks. Future work could include multi-task prompt-based pre-training methods (Sanh et al., 2021; Wei et al., 2021).

Furthermore, we observe noisy examples in synthetic dataset on difficult tasks such as NLI and QA, this situation progressively deteriorates when incorporating more diverse decoding strategy (e.g., Nucleus Sampling). Better decoding strategies are needed to ensure the label correctness while preserving the dataset diversity (Massarelli et al., 2020). Besides, methods that learn from noisy labels can be integrated into the training of the tiny task model (Song et al., 2020).

We hope this paper can provide contributions for further exploiting dataset-generation-based zero-shot learning with large pre-trained language models.

Limitations

Although ZEROGEN achieves promising performance under zero-shot learning setting, this choice does come with certain limitations. We find the stability regarding the prompt choice of ZEROGEN is still far from satisfactory. ZEROGEN underperforms PROMPTING in some certain selected prompts, and prompt engineering is tough as it's shown a different preference on prompts across various tasks. Future work may include multi-task prompt-based pre-training methods (Sanh et al., 2021; Wei et al., 2021) to improve prompt robustness.

We also observe noisy examples in the synthetic dataset on difficult tasks such as NLI and QA, this situation progressively deteriorates when incorporating a more diverse decoding strategy (e.g., Nucleus Sampling). Better decoding strategies are needed to ensure the label's correctness while preserving the dataset diversity (Massarelli et al., 2020). Reciprocally, methods that learn from noisy labels can be integrated into the training of the tiny task model (Song et al., 2020).

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.

Max Bartolo, Alastair Roberts, Johannes Welbl, Sebastian Riedel, and Pontus Stenetorp. 2020. Beat the AI: investigating adversarial human annotation for reading comprehension. *Trans. Assoc. Comput. Linguistics*, 8:662–678.

Luisa Bentivogli, Bernardo Magnini, Ido Dagan, Hoa Trang Dang, and Danilo Giampiccolo. 2009. The fifth PASCAL recognizing textual entailment challenge. In *Proceedings of the Second Text Analysis Conference, TAC 2009, Gaithersburg, Maryland, USA, November 16-17, 2009*. NIST.

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*.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The PASCAL recognising textual entailment challenge. In *Machine Learning Challenges, Evaluating Predictive Uncertainty, Visual Object Classification and Recognizing Textual Entailment, First PASCAL Machine Learning Challenges Workshop, MLCW 2005, Southampton, UK, April 11-13, 2005, Revised Selected Papers*, volume 3944 of *Lecture Notes in Computer Science*, pages 177–190. Springer.

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.

William Fedus, Barret Zoph, and Noam Shazeer. 2021. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *CoRR*, abs/2101.03961.

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. 2019. Mask-predict: Parallel decoding of conditional masked language models. 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 6111–6120. Association for Computational Linguistics.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and Bill Dolan. 2007. The third PASCAL recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL@ACL 2007 Workshop on Textual Entailment and Paraphrasing, Prague, Czech Republic, June 28-29, 2007*, pages 1–9. Association for Computational Linguistics.

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

Mitchell A. Gordon, Kevin Duh, and Nicholas Andrews. 2020. Compressing BERT: studying the effects of weight pruning on transfer learning. In *Proceedings of the 5th Workshop on Representation Learning for NLP, RepL4NLP@ACL 2020, Online, July 9, 2020*, pages 143–155. Association for Computational Linguistics.

Yves Grandvalet and Yoshua Bengio. 2006. Entropy regularization.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor OK Li, and Richard Socher. 2018. Non-autoregressive neural machine translation. In *International Conference on Learning Representations*.

Jian Guan and Minlie Huang. 2020. UNION: an unreferenced metric for evaluating open-ended story generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 9157–9166. Association for Computational Linguistics.

R Bar Haim, Ido Dagan, Bill Dolan, Lisa Ferro, Danilo Giampiccolo, Bernardo Magnini, and Idan Szpektor. 2006. The second pascal recognising textual entailment challenge. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*.

Xuanli He, Islam Nassar, Jamie Kiros, Gholamreza Haffari, and Mohammad Norouzi. 2021. Generate, annotate, and learn: Generative models advance self-training and knowledge distillation. *CoRR*, abs/2106.06168.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

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.

Paras Jain, Xiangxi Mo, Ajay Jain, Alexey Tumanov, Joseph E. Gonzalez, and Ion Stoica. 2019. The ooo VLIW JIT compiler for GPU inference. *CoRR*, abs/1901.10008.

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.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. Tinybert: Distilling BERT for natural language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 4163–4174. Association for Computational Linguistics.

Nitish Shirish Keskar, Bryan McCann, Lav Varshney, Caiming Xiong, and Richard Socher. 2019. CTRL - A Conditional Transformer Language Model for Controllable Generation. *arXiv preprint arXiv:1909.05858*.

Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. Unifiedqa: Crossing format boundaries with a single QA system. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 1896–1907. Association for Computational Linguistics.

Sehoon Kim, Amir Gholami, Zhewei Yao, Michael W Mahoney, and Kurt Keutzer. 2021. I-bert: Integer-only bert quantization. *arXiv preprint arXiv:2101.01321*.

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*.

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

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

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.

Jiwei Li, Will Monroe, and Dan Jurafsky. 2016. A simple, fast diverse decoding algorithm for neural generation. *CoRR*, abs/1611.08562.

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

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. *CoRR*, abs/1907.11692.

Yuang Liu, Wei Zhang, Jun Wang, and Jianyong Wang. 2021. Data-free knowledge transfer: A survey. *CoRR*, abs/2112.15278.

Xindian Ma, Peng Zhang, Shuai Zhang, Nan Duan, Yuexian Hou, Ming Zhou, and Dawei Song. 2019. A tensorized transformer for language modeling. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 2229–2239.

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.

Luca Massarelli, Fabio Petroni, Aleksandra Piktus, Myle Ott, Tim Rocktäschel, Vassilis Plachouras, Fabrizio Silvestri, and Sebastian Riedel. 2020. How decoding strategies affect the verifiability of generated text. In *Findings of the Association for*

Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020, volume EMNLP 2020 of *Findings of ACL*, pages 223–235. Association for Computational Linguistics.

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.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, Yejin Choi, and Hannaneh Hajishirzi. 2021a. Reframing instructional prompts to gptk's language. *arXiv preprint arXiv:2109.07830*.

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

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2021c. Natural instructions: Benchmarking generalization to new tasks from natural language instructions. *CoRR*, abs/2104.08773.

Matan Ben Noach and Yoav Goldberg. 2020. Compressing pre-trained language models by matrix decomposition. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing, AACL/IJCNLP 2020, Suzhou, China, December 4-7, 2020*, pages 884–889. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1532–1543. ACL.

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 Zaid Harchaoui. 2021. Mauve: Measuring the gap between neural text and human text using divergence frontiers. *Advances in Neural Information Processing Systems*, 34.

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

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100, 000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 2383–2392. The Association for Computational Linguistics.

Ahmad Rashid, Vasileios Lioutas, Abbas Ghaddar, and Mehdi Rezagholizadeh. 2021. Towards zero-shot knowledge distillation for natural language processing. 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 6551–6561. Association for Computational Linguistics.

Laria Reynolds and Kyle McDonell. 2021. Prompt programming for large language models: Beyond the few-shot paradigm. In *CHI '21: CHI Conference on Human Factors in Computing Systems, Virtual Event / Yokohama Japan, May 8-13, 2021, Extended Abstracts*, pages 314:1–314:7. ACM.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, Manan Dey, M. Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal V. Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Févry, Jason Alan Fries, Ryan Teehan, Stella Biderman, Leo Gao, Tali Bers, Thomas Wolf, and Alexander M. Rush. 2021. Multitask prompted training enables zero-shot task generalization. *CoRR*, abs/2110.08207.

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.

Timo Schick, Sahana Udupa, and Hinrich Schütze. 2021. Self-diagnosis and self-debiasing: A proposal for reducing corpus-based bias in NLP. *CoRR*, abs/2103.00453.

Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Bidirectional attention flow for machine comprehension. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Sheng Shen, Zhen Dong, Jiayu Ye, Linjian Ma, Zhewei Yao, Amir Gholami, Michael W. Mahoney, and Kurt Keutzer. 2020. Q-BERT: hessian based ultra low precision quantization of BERT. 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 8815–8821. AAAI Press.

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.

Hwanjun Song, Minseok Kim, Dongmin Park, and Jae-Gil Lee. 2020. Learning from noisy labels with deep neural networks: A survey. *CoRR*, abs/2007.08199.

Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, and Denny Zhou. 2020. Mobilebert: a compact task-agnostic BERT for resource-limited devices. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 2158–2170. Association for Computational Linguistics.

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.

Ashwin K. Vijayakumar, Michael Cogswell, Ramprasaath R. Selvaraju, Qing Sun, Stefan Lee, David J. Crandall, and Dhruv Batra. 2016. Diverse beam search: Decoding diverse solutions from neural sequence models. *CoRR*, abs/1610.02424.

Tu Vu, Minh-Thang Luong, Quoc V. Le, Grady Simon, and Mohit Iyyer. 2021. Strata: Self-training with task augmentation for better few-shot learning. 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 5715–5731. Association for Computational Linguistics.

Ziheng Wang, Jeremy Wohlwend, and Tao Lei. 2020. Structured pruning of large language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 6151–6162. Association for Computational Linguistics.

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

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2021. Finetuned language models are zero-shot learners. *CoRR*, abs/2109.01652.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2020. Neural text generation with unlikelihood training. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

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.

Hanwei Xu, Yujun Chen, Yulun Du, Nan Shao, Yanggang Wang, Haiyu Li, and Zhilin Yang. 2022. Zeroprompt: Scaling prompt-based pretraining to 1,000 tasks improves zero-shot generalization. *CoRR*, abs/2201.06910.

Jingjing Xu, Wangchunshu Zhou, Zhiyi Fu, Hao Zhou, and Lei Li. 2021. A survey on green deep learning. *CoRR*, abs/2111.05193.

Jiacheng Ye, Tao Gui, Yichao Luo, Yige Xu, and Qi Zhang. 2021. One2set: Generating diverse keyphrases as a set. 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 4598–4608. Association for Computational Linguistics.

Ofir Zafrir, Guy Boudoukh, Peter Izsak, and Moshe Wasserblat. 2019. Q8BERT: quantized 8bit BERT. In *Fifth Workshop on Energy Efficient Machine Learning and Cognitive Computing - NeurIPS Edition, EMC2@NeurIPS 2019, Vancouver, Canada, December 13, 2019*, pages 36–39. IEEE.

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

Ruiqi Zhong, Kristy Lee, Zheng Zhang, and Dan Klein. 2021. Adapting language models for zero-shot learning by meta-tuning on dataset and prompt collections. In *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*, pages 2856–2878. Association for Computational Linguistics.

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 Experimental Setup

Implementation Details For dataset generation, we use Nucleus Sampling (Holtzman et al., 2020) with $p = 0.9$ by default as it is considered to be able to generate both fluent and diverse texts(Holtzman et al., 2020). The scale of synthetic dataset is 200k in the main results, and 100k in other analysis experiments. Regarding prompt selection, we manually design a series of prompts for each task, and report results on the best prompt for PROMPTING and ZEROGEN framework. For NLI tasks, we adopt self-debiasing mechanism with a decay constant of 200 (Schick et al., 2021) to ensure that each generated text pair is not only a good fit for a given label, but also not a good fit for other labels (Schick and Schütze, 2021). We removing overly short/long sentences, and sentences without an ending quotation mark.

We implement a LSTM-based model and a DistilBERT model as TAM. For LSTM-based model, we use Adam optimizer (Kingma and Ba, 2015), a learning rate of 1e-4, an embedding dim of 100, and a hidden size of 300. For single sentence classification(TC), we use 1-layer BiLTSM to encode the sentence and use a linear classifier. For sentence-pair classification(NLI), we use 2-layer BiLTSM to encode the sentences to v_1, v_2 and pass the concatenated $[v_1; v_2; |v_1 - v_2|; v_1 * v_2]$ to a classifier. For QA tasks, we use the 1-layer BiDAF model. To ensure that TAMs are truly trained from scratch using the synthetic corpus, we random initialize TAMs' embedding without using any pre-trained word embeddings (e.g., GloVe (Pennington et al., 2014). 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). We run experiments on a single NVIDIA A100 80G GPU, and generating 200k examples cost 12h on average.

B Additional Results on Prompt Design

For Question Answering tasks, the *natural language style* prompt is also the most suitable for both PROMPTING and ZEROGEN settings, achieving the highest scores. However, for Natural Language Inference tasks, the most suitable prompts for QNLI and RTE are different. For RTE, the *natural language style* prompt is best, while the *control code* prompts perform significantly better than *natural language style* prompts in QNLI.

C Additional Related Work on Efficient Inference of PLMs

There is a line of works dedicated to improving the inference efficiency of PLMs, including pruning (Wang et al., 2020; Gordon et al., 2020), low-rank factorization (Ma et al., 2019; Noach and Goldberg, 2020; Lan et al., 2020), quantization (Zafrir et al., 2019; Shen et al., 2020; Kim et al., 2021), knowledge distillation (Jiao et al., 2020; Sanh et al., 2019; Sun et al., 2020) and parallel decoding (Gu et al., 2018; Ghazvininejad et al., 2019; Ye et al., 2021). We refer the readers to Xu et al. (2021) for a detailed survey. Concerning privacy, copyright or confidentiality, data-free knowledge distillation (DFKD) (Liu et al., 2021) has attracted appealing attention in computer vision field as it deals with distilling valuable knowledge from well-trained models without requiring to access to the training data. However, similar approaches for NLP are difficult to work due to discrete character of words. Rashid et al. (2021) relax the data-free condition and use out-of-distribution labeled data to train a generator. By contrast, our method generates data with the PLMs (i.e., the teacher), without requiring any pre-defined labeled data. In the literature of knowledge

distillation, ZEROGEN framework could produce a student model that achieves superior zero-shot performance the teacher model.

D ZEROGEN as Knowledge Distillation

ZEROGEN can be seen as a dataset-based knowledge distillation framework. We compare vanilla knowledge distillation baselines with ZEROGEN in Table 7. The soft and hard labels are generated by GPT2-XL on the unlabeled training set. The generated labels are used to train a tiny task model for comparison. The superior results on ZEROGEN show that the paradigm can better utilize large PLM by distilling more knowledge into a large amount of input-output pairs, while vanilla knowledge distillation purely distills knowledge into outputs.

TAMs	Supervised	KD-HARD	KD-SOFT	ZEROGEN
LSTM	84.60	75.23	68.31	79.80
DistilBERT	87.24	82.32	80.21	84.28

Table 7: Comparison with knowledge distillation (KD) baselines on IMDb. KD-HARD and KD-SOFT represent KD baselines using hard labels and soft labels, respectively.

E ZEROGEN for Data Augmentation

We report the results using the synthetic data as augmentation data in Tabel 8. The results show that the zero-shot synthetic data is a good supplement to human-annotated data (gold data) and can improve the model performance.

Data	Sample Size	LSTM	DistilBERT
Gold	25,000	84.60	87.24
Gold + AUG-200k	225,000	88.91	93.42
Gold + AUG-500k	525,000	90.42	93.59

Table 8: Results of data augmentation on IMDb using synthetic data. AUG-200k and AUG-500k represent using 200k and 500k synthetic data respectively.

F ZEROGEN for Self-improving

We have shown that a tiny task model can outperform a large PLM after training on the synthetic dataset. A natural question is *"Can PLM improve its own performance after tuning on the dataset generated by itself?"*. We experiment using PLM as TAM and report the results in Table 9. To summarize, we find 1) A larger TAM can further boosts the performance; 2) PLMs can improve itself by fine-tuning on the dataset generated by its own.

PLMs	TAMs	IMDb	SQuAD	QNLI
-	LSTM	84.60	41.86/57.22	69.00
	DistilBERT	87.24	76.28/84.67	88.05
	GPT2-XL	95.68	76.92/85.48	92.88
GPT2-XL	-	80.64	4.61/13.32	60.60
	LSTM	79.80	12.35/18.66	51.53
	DistilBERT	84.28	25.50/31.53	71.19
	GPT2-XL	**90.71**	**25.78/32.13**	**73.69**

Table 9: Results of PLM (i.e., GPT2-XL) fine-tuned with gold (upper) and synthetic (lower) dataset. A larger TAM further boosts the performance, and PLM can improve its own performance after fine-tuning on synthetic dataset by itself (grey blocks).

G Generated Examples

We present some qualitative examples for different tasks in Appendix Table 11. Text classification task (SST-2) is relatively simple and concise, the generated samples generally fit the prompts and sentiment polarity well by using descriptive tokens about the given movie name and positive/negative sentiment. Take the first case in SST-2 as an example, the generated tokens "*action-adventure*" and "*attractive*" are the natural continuations for movie name "*The Spiderwick Chronicles (Movie)*" and "*positive*" sentiment in prompt. Although natural language inference tasks are complex, the generated questions (QNLI) and inferences (RTE) could respond to different types of prompts and relate to the given contexts (e.g., the generated question drifts topic for prompt "*Information:... Question (answer **not in** above information)*" in QNLI). While the context of the question answering task (SQuAD) is long and contains a lot of information, ZEROGEN could successfully generate question "*Who is the one and only true God ?*" which is used to response to the pre-set answer "*Jehovah*". Overall, these generation examples show that ZEROGEN can generate useful and arbitrary number of training samples that could be used to train TAMs.

Setting	Id	Prompt	Prompt Type	SQuAD	AdversarialQA
PROMPTING	P_1	Context: "<C>"\nQuestion: "<X>"\nAnswer: "	Control code	2.69/10.90	1.33/5.61
	P_2	Task: Generate an answer given the context and question. \nContext: "<C>"\nQuestion: "<X>"\nAnswer: "	Control code with task description	3.81/12.1	1.60/5.89
	P_3	"<C>"\n The answer to the question "<X>" is: "	Natural language style	4.41/12.60	2.00/5.78
	P_4	"<C>"\nBased on the above description, the answer to the question "<X>" is: "		6.33/15.49	2.27/6.71
	P_5	The context is: "<C>"\nThe answer to the question "<X>" is: "		4.61/13.32	2.13/6.30
ZEROGEN	$P_1^{'}$	Context: "<C>"\nAnswer: "<Y>"\nQuestion: "	Control code	9.96/9.62	2.87/5.33
	$P_2^{'}$	Task: Generate a question given the context and answer. \nContext: "<C>"\nAnswer: "<Y>"\nQuestion: "	Control code with task description	5.84/8.70	2.77/5.21
	$P_3^{'}$	"<C>"\n"<Y>" is the answer to the question: "	Natural language style	24.55/29.36	5.30/8.82
	$P_4^{'}$	"<C>"\nBased on the above description, "<Y>" is the answer to the question: "		23.58/29.84	5.87/9.58
	$P_5^{'}$	The context is: "<C>"\n"<Y>" is the answer to the following question: "		23.93/30.62	5.97/10.02

Setting	Id	Prompt	Label words <Y>	Prompt Type	QNLI
PROMPTING	P_1	Context: "<X>"\nQuestion (answer <Y> the context): "	in/not in	Control code	50.51
	P_2	Information: "<X>"\nQuestion (answer <Y> above information): "	in/not in		50.52
	P_3	"<X>"\n Based on the above description, the following question is [Y1] and [Y2] be answered: "	clear/not clear can/can not	Natural language style	60.42
	P_4	The context sentence is: "<X>"\nThe question is: "<X>"\n The context sentence <Y> the answer to the question.	contains/ doesn't contain		60.61
	P_5	The context sentence is: "<X>"\nThe context sentence <Y> the answer to the following question.\nThe question is: "<X>"	contains/ doesn't contain		58.02
ZEROGEN	P_1	Context: "<X>"\nQuestion (answer <Y> the context): "	in/not in	Control code	69.82
	P_2	Information: "<X>"\nQuestion (answer <Y> above information): "	in/not in		70.55
	P_3	"<X>"\n Based on the above description, the following question is [Y1] and [Y2] be answered: "	clear/not clear can/can not	Natural language style	53.71
	$P_4^{'}$	The context sentence is: "<X>"\nThe context sentence <Y> the answer to the following question.\nThe question is: "<X>"	contains/ doesn't contain		55.65

Setting	Id	Prompt	Label words <Y>	Prompt Type	RTE
PROMPTING	P_1	Task: Write two sentences have the <Y> meaning. \nThe first sentence: "<X>"\nThe second sentence: "	same/complete different	Control code with task description	52.71
	P_2	Based on the fact that "<X>", it is <Y> that: "	correct/not correct	Natural language style	51.26
	P_3	Suppose "<X>", we <Y> infer that: "	can/ can not		52.71
	P_4	The context is: "<X>"\nBased on the context, it is <Y> that: "	correct/not correct		54.51
	P_5	"<X>"\nBased on the above description, the following sentence is definitely <Y> : "	correct/wrong		57.04
ZEROGEN	P_1	Task: Write two sentences have the <Y> meaning. \nThe first sentence: "<X>"\nThe second sentence: "	same/complete different	Control code with task description	47.29
	P_2	Based on the fact that "<X>", it is <Y> that: "	correct/not correct	Natural language style	57.76
	P_3	Suppose "<X>", we <Y> infer that: "	can/ can not		49.46
	P_4	The context is: "<X>"\nBased on the context, it is <Y> that: "	correct/not correct		61.73
	P_5	"<X>"\nBased on the above description, the following sentence is definitely <Y> : "	correct/wrong		62.82

Table 10: Results for different prompts on Question Answering and Natural language Inference tasks. Results are reported on dev set. For Question Answering, "<C>" represents the input context, "<X>" represents the input question, "<Y>" represents the generated answer. "\n" represents the newline symbol. $P^{'}$ represents minor revised version of P for text generation.

Task	Prompt	Sampled Generation
SST-2	The movie review in **positive** sentiment for movie "**The Spiderwick Chronicles (Movie)**" is: "	It's an action-adventure movie about a crime fighter who must face an even bigger foe, then the end of the world, all in a funny, attractive, and moving way.
	The movie review in **negative** sentiment for movie "**Fist Fight**" is: "	The action scenes are laughable, the characters lack personality, and the plot is boring.
QNLI	Information: "**In September 1695, Captain Henry Every, an English pirate on board the Fancy, reached the Straits of Bab-el-Mandeb, where he teamed up with five other pirate captains to make an attack on the Indian fleet making the annual voyage to Mocha.**" \n Question (answer **in** above information): "	Why did each of these captains enter the Straits of Bab-el-Mandeb?
	Information: "**Gaddafi remained the government's public face, with the identities of the other RCC members only being publicly revealed on 10 January 1970.**" \n Question (answer **not in** above information): "	Was the financing received from the international revolution from Germany (4)?
RTE	"**Today about 75% of people live in cities or towns. Art and culture are an important part of France.**" \n Based on the above description, the following sentence is definitely **correct**: "	Today about 75% of people live in cities and towns.
	"**There are four identified subtypes of Ebola virus. Three of the four have caused disease in humans: Ebola-Zaire, Ebola-Sudan, and Ebola-Ivory Coast. The fourth, Ebola-Reston, has caused disease in non-human primates, but not in humans.**" \n Based on the above description, the following sentence is definitely **wrong**: "	The most recently documented case was a Sierra Leonean man who died in 2015.
SQuAD	The context is: "**On the other hand, in his study on nine of the Bibles most widely in use in the English-speaking world, Bible scholar Jason BeDuhn, Professor of Religious Studies at the Northern Arizona University, wrote: Ž01cThe NW [New World Translation] emerges as the most accurate of the translations compared.Ž01d Although the general public and many Bible scholars assume that the differences in the New World Translation are the result of religious bias on the part of its translators, BeDuhn stated: Ž01cMost of the differences are due to the greater accuracy of the NW as a literal, conservative translation of the original expressions of the New Testament writers.Ž01d He added however that the insertion of the name Jehovah in the New Testament violate[s] accuracy in favor of denominationally preferred expressions for God.**" \n "**Jehovah**" is the answer to the following question: "	Who is the one and only true God ?

Table 11: Generated examples for each task. We omit the example for IMDb and Adversarial QA tasks since we use the exactly same prompt as SST-2 and SQuAD task, respectively. The input conditions in each prompt are **bolded**.