SELF-CONSISTENT LEARNING: COOPERATION BETWEEN GENERATORS AND DISCRIMINATORS

Anonymous authors
Paper under double-blind review

ABSTRACT

Using generated data to improve the performance of downstream discriminative models has recently gained popularity due to the great development of pre-trained language models. In most previous studies, generative models and discriminative models are trained separately, thus could not adapt to any changes in each other. As a result, the generated samples can easily deviate from the real data distribution, while the improvement of the discriminative model quickly reaches saturation. Generative adversarial networks (GANs) train generative models via an adversarial process with discriminative models to achieve joint training. However, the training of standard GANs is notoriously unstable and often falls short of convergence. In this paper, to address these issues, we propose a *self-consistent learning* framework, in which a discriminator and a generator are cooperatively trained in a closed-loop form. The discriminator and the generator enhance each other during multiple rounds of alternating training until a scoring consensus is reached. This framework proves to be easy to train and free from instabilities such as mode collapse and non-convergence. Extensive experiments on sentence semantic matching demonstrate the effectiveness of the proposed framework: the discriminator achieves 10+ AP of improvement on zero-shot setting and new state-of-the-art performance on full-data setting.

1 INTRODUCTION

The advance of Pre-trained Language Models (PLMs) (Raffel et al., 2020; Brown et al., 2020; Chowdhery et al., 2022) has substantially improved the performance of deep neural networks across a variety of Natural Language Processing (NLP) tasks. Various language models, based on Transformer (Vaswani et al., 2017) architecture, have been proposed, leading to state-of-the-art (SOTA) performance on the fundamental discrimination tasks. These models are first trained with self-supervised training objectives (for instance predicting masked tokens according to surrounding tokens) on massive unlabeled text data, then fine-tuned on annotated data to adapt to downstream tasks of interest. However, annotated data is usually limited to a wide range of downstream tasks, which results in overfitting and a lack of generalization to unseen data.

One straightforward way to deal with such data scarcity problem is data augmentation (Xie et al., 2020), and incorporating generative models has been widely adopted recently (Gangal et al., 2022; Carlini et al., 2021; Wang et al., 2021). Despite its popularity, the generated text can easily deviate from the real data distribution without exploiting any of the signals passed back from the discrimination task. In previous studies, generative data augmentation and discrimination have been well studied as separate problems, but it is less clear how the two can be leveraged in one framework and their performances can be improved simultaneously.

Generative Adversarial Networks (GANs) (Goodfellow et al., 2014; Gulrajani et al., 2017) are good attempt to couple generative models and discriminative models in an adversarial manner, where a two-player minimax game between learners are carefully crafted. GANs have achieved tremendous success in domains such as image generation (Denton et al., 2015), and related studies have also shown their effectiveness in semi-supervised learning (Salimans et al., 2016; Kumar et al., 2017). However, GANs are notoriously difficult to train, most training objectives only work well for one model, either the discriminator or the generator, so rarely both learners can be optimal at the same time (Arjovsky & Bottou, 2017; Wiatrak et al., 2019). This essentially arises from the adversarial

nature of GANs, that during the process, optimizing one learner can easily destroy the learning ability of the other, making GANs fail to converge.

Another limitation of simultaneously optimizing the generator and the discriminator models comes from the discrete nature of text in NLP, as no gradient propagation can be done from the discriminators to the generators. One theoretically sound attempt is to use reinforcement learning (RL), but the sparsity and the high variance of the rewards make the training in NLP tasks particularly unstable.

To address these shortcomings, we novelly introduce a self-consistent learning framework based on one generator and one discriminator: the generator and the discriminator is alternately trained by way of cooperation rather than adversary. Specifically, in each round of training, the samples generated by the generator are synthetically labeled by the discriminator, and then only part of them would be selected based on dynamic thresholds and used for the training of the discriminator and the generator in the next round. Several benefits can be uncovered from this cooperative training process. First, a closed-loop form of cooperation can be established so that we can get the optimal generator and discriminator at the same time. Second, this framework helps improve the generation quality while ensuring the generator's domain specificity, which in turn contributes to the training. Third, a steady stream of diverse synthetic samples can be added to the training in each round and lead to continuous improvement of the performance of all learners. Finally, we can start the training with only domain-related corpus and obtain strong results, while these data can be easily sampled without additional cost or supervision. Furthermore, the performance on labeled datasets can be further boosted on the basis of the SOTA level. As an example to demonstrate the effectiveness of our framework, We examine it on the task of sentence semantic matching. The experiments show that our method significantly improves over standalone state-of-the-art discriminative models on zero-shot and full-data settings.

Our contributions are summarized as follows,

- We propose a self-consistent learning framework that incorporates the generator and the discriminator, in which remarkable gains in the performance of both will be achieved simultaneously.
- We propose a dynamic selection mechanism such that the cooperation between the generator and the discriminator drives the convergence to reach their scoring consensus.
- Experimental results show that our proposed framework significantly outperforms state-of-the-art methods for the task of sentence semantic matching.

2 Related Works

To alleviate the lack of annotated data in supervised learning in NLP, semi-supervised learning (SSL) has been a popular line of research (Van Engelen & Hoos, 2020). In terms of the sources of the unlabeled data required by SSL, they are either collected from the domains or recently generated by generative language models. Then NLU models can learn from the unlabeled data by pseudo-labeling (Arazo et al., 2020; Banitalebi-Dehkordi & Zhang, 2021) and consistent regularization (Jeong et al., 2019; Sohn et al., 2020). However, the collection of unlabeled data comes at a cost, and even with generative models, there is no guarantee of the quality of the generated samples because the model cannot tune the generation based on the performance of the downstream tasks. In contrast, our method always includes a continuously updated generative model, which dynamically adjusts its generation according to the performance of downstream tasks.

Previous work borrows the idea of GANs in image generation to NLP. In this framework, the generator is adversarially trained with the discriminator. Unlike conventional GANs in continuous domains, language GANs usually employ Gumbel-Softmax differentiation (Jang et al., 2017; Yin et al., 2020), Reinforcement Learning (RL) (Yu et al., 2017; Wu et al., 2021), or modified training objectives (Che et al., 2017; Montahaei et al., 2021) to update the generator, in order to leverage the non-differential signals from the discriminator. However, language GANs are often criticized for underperforming Maximum likelihood estimation (MLE) and are very difficult to train, even the single optimality of either the generator or the discriminator cannot be guaranteed (Alvarez-Melis et al., 2022). On the other hand, our proposed framework allows the incorporation of the generator and the discriminator in a cooperative way, leading to continuous improvement of both learners and outperforming SOTA.

3 METHODOLOGY

3.1 SELF-CONSISTENT LEARNING FRAMEWORK

In this section, we introduce our semi-supervised self-consistent learning (**SCL**) framework.

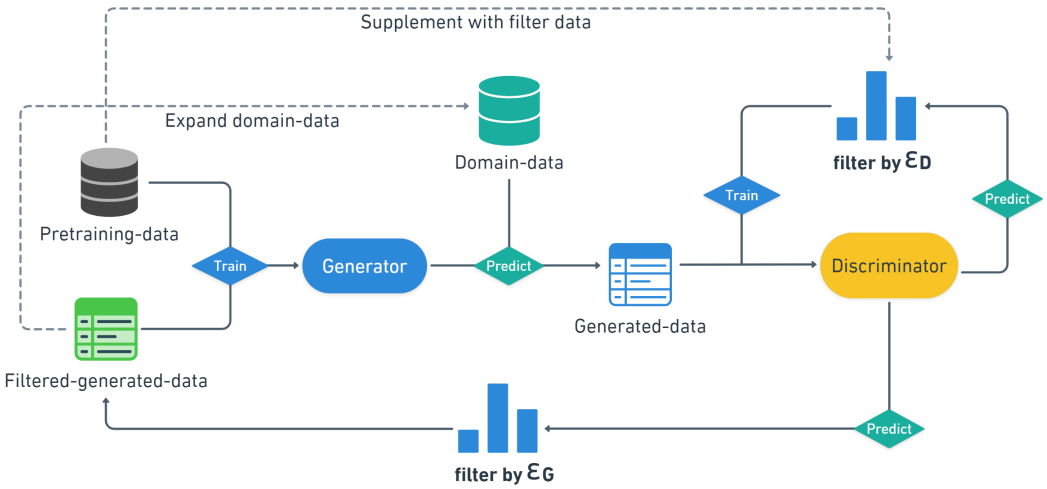

Figure 1: Overall flow chart of our SCL framework.

As shown in Figure 1, our framework, similar to the GANs, consists of a generator and a discriminator model. However, contrasting to the GANs, these two parts in our framework work cooperatively to enhance each other. Specifically, for any given class k, the generator $s_k^b = \mathcal{G}(s_k^a)$ will now become a conditional generator that takes in an input sentence s_k^a and generate an output sentence s_k^b. The discriminator $\mathcal{D}(s_k^b)$ is then responsible for discriminating the sentence by using a dynamic threshold $\epsilon_{\mathcal{D}}$, and will use the discriminated sentence as either positive or negative data points for that specific class to continue its training process. Once the new discriminator is trained, the sentence will be discriminated again by the new discriminator with a different dynamic threshold $\epsilon_{\mathcal{G}}$. This time only the positive cases will be passed to the generator as the training data for the new round, thus completing a close loop of cooperation.

Theorem 1. At round t, the aim of the optimization of generator $\mathcal{G}_\theta^t$, given the previous round discriminator $\mathcal{D}_\phi^{t-1}$, boils down to,

$$\min_\theta \mathbb{D}_{KL}(p_{\mathcal{D}_\phi^{t-1}}(\cdot), p_{\mathcal{G}_\theta^t}(\cdot)) \tag{1}$$

where $\mathbb{D}_{KL}$ is the standard KL divergence.

Theorem 1 shows that the generator at round t is encouraged to approximate the score distribution given by the previous round discriminator measured by KL divergence. In particular, based on a well-pretrained discriminator, the generation is correspondingly guaranteed to be faithful to the real data distribution. (proof given in Appendix A)

Throughout the process, the selection mechanism keeps samples whose output probability $p_{\mathcal{D}}$ is greater than the threshold ϵ, while other generated samples with confidence lower than the threshold ϵ will be discarded:

$$p_{\mathcal{D}}(y = k|s_k^b) = \text{softmax}(\text{MLP}(\mathbf{H})) \tag{2}$$

$$filter(y) \triangleq p_{\mathcal{D}}(y = k|s_k^b) \geq \epsilon_{t,k} \tag{3}$$

$$\epsilon_{t,k} = f(t, \mathcal{L}_{t-1,k}^*, \epsilon_{t-1,k}^*),\ where\, \mathcal{L}_{0,k}^* = 0, \epsilon_{0,k}^* = \lambda \tag{4}$$

In Equation 2, $\mathbf{H}$ represents the embedding representation of $\{s_k^b\}$, $p_{\mathcal{D}}(y = k|s_k^b)$ represents the probability that $\{s_k^b\}$ belongs to class k. In Equation 3, $\epsilon_{t,k}$ represents the dynamic threshold for accepting $\{s_k^b\}$ as negative or positive samples in this round. In Equation 4, the threshold $\epsilon_{t,k}$ is a function about the current round number t, the discriminator loss $\mathcal{L}^*_{t-1,k}$ and the $\epsilon^*_{t-1,k}$ from last round (where λ is a hyper-parameter). Note this function serves a critical role in closing the gap between the generator and discriminator and is the key points why this cooperation loops forms a virtuous circle[1].

The reasons that this framework is cooperative instead of adversarial are: (1) the generator is no longer a challenger to the discriminator that only provide negative data points to fool it, but now serves as a data augmenter to provide both positive and negative data points to enhance the discriminator; (2) the generator no longer updates its parameters through the policy gradients guided by the signals from the discriminator, but rather by utilizing the filtered data points to further improve its conditional generation quality. Note that by deliberately choosing the conditional generation paradigm along with the selection mechanism, we not only make the training more stable due to the different training goal, but also bypass the mode collapse problem of GANs(see Section 4.6 for further discussion). Besides, by iterating through the loops, our framework achieves self-consistency by honing the generator's domain specificity and increasing the discriminator's domain data exposure.

3.2 Sentence Semantic Matching

We leverage the sentence semantic matching task as an example to demonstrate the effectiveness of our method in binary classification setting(e.g. $|k| = 2$). First, let's introduce the formal definition of this task. Given two sentences $s^a = \{w_1^a, w_2^a, ..., w_{l_a}^a\}$ and $s^b = \{w_1^b, w_2^b, ..., w_{l_b}^b\}$, where w_i^a and w_j^b represents the i-th and j-th tokens in the sentences, and l_a and l_b indicate the total length of s^a and s^b. The goal of this task is to learn a discriminator $\mathcal{D}$ which is able to precisely predict the label $y = \mathcal{D}(s^a, s^b)$, where $y \in \mathcal{Y} = \{0, 1\}$.

In our task, $\mathcal{G}$ is trained to generate a similar sentence s^b from any given sentence s^a and $\mathcal{D}$ is trained to predict label y from any given sentence pair $\{s^a, s^b\}$. As demonstrated in Figure 1, there are mainly two training processes in the whole framework: fix $\mathcal{G}$ to train $\mathcal{D}$ and fix $\mathcal{D}$ to train $\mathcal{G}$.

Training $\mathcal{D}$: We first randomly sample s^a from domain-related corpus, and then input s^a to $\mathcal{G}$ to generate s^b. Next, we feed sentence pair $\{s^a, s^b\}$ into $\mathcal{D}_{old}$ to predict the label y_{old}, and filter $\{s^a, s^b, y_{old}\}$ using threshold $\epsilon_{\mathcal{D}}$. Finally we train $\mathcal{D}_{old}$ on the selected data and pretraining data P to get an improved discriminator $\mathcal{D}_{new}$. The update process seeks to minimize the cross-entropy loss over all instances:

$$\mathcal{L}_{\mathcal{D}}(\mathbf{s}, \mathbf{y}) = \frac{1}{|\mathbf{s}|} \sum_{i=1}^{|\mathbf{s}|} -[y_i \cdot \log p_{\mathcal{D}}(y_i = 1|s_i^a, s_i^b) + (1 - y_i) \cdot \log(1 - p_{\mathcal{D}}(y_i = 1|s_i^a, s_i^b))] \quad (5)$$

Training $\mathcal{G}$: We feed the generated sentence pairs $\{s^a, s^b\}$ into $\mathcal{D}_{new}$ to predict new labels y_{new}, and then filter $\{s^a, s^b, y_{new}\}$ using threshold $\epsilon_{\mathcal{G}}$ and additional rules [2]. For the filtered data, we supplement it with the pretraining data P to update $\mathcal{G}_{old}$ to $\mathcal{G}_{new}$ [3]. We also take out the s^b from the filtered data and add them to domain-related corpus. The expanded domain corpus will be used to sample conditional sentences in the next round of generation. The update procedure of $\mathcal{G}$ employs the negative log-likelihood function over all instances:

$$\mathcal{L}_{\mathcal{G}}(\mathbf{s^a}, \mathbf{s^b}) = -\frac{1}{|\mathbf{s^b}|} \sum_{t=1}^{|\mathbf{s^b}|} \log p_{\mathcal{G}_\theta}(s_t^b|s_{<t}^b, \mathbf{s^a}) \quad (6)$$

[1] ϵ_D and ϵ_G are different thresholds determined by equation 4 separately

[2] The additional rule aims to remove those sentences that are too long, too short, or too identical according to the longest common substring algorithm.

[3] Note that the pretraining data P are used to pretrain $\mathcal{G}$ and $\mathcal{D}$ in the warm-up phase. Theoretically, the pretraining data are not mandatory but we empirically found it can alleviate the domain shifting problem and prevent the generation distribution from drifting.

For the selection mechanism, we used the form of $\epsilon_t = k * t + \lambda$ after comparing the effects of different functions through experiments according to Equation 4. where k is the increment of the threshold for each round, t is the number of training rounds, λ is the initial threshold, ϵ_t is the threshold of t-th rounds.

In the process of training $\mathcal{G}$, since the sentences generated in each round will be added to the domain-related corpus, the source of domain specific data is thus monotonically expanding by iterating the self-consistent learning loop. The more formalized process is shown in Algorithm 1.

Algorithm 1 Self-consistent Learning (**SCL**)

Require: Generator $\mathcal{G}$; Discriminator $\mathcal{D}$; Domain-Related Corpus C; Pretraining Data P.

```
Initialize G_0 and D_0 with pre-trained language models;
Warm up G_0 and D_0 with pretraining data P to get G_1 and D_1;
for each i ∈ [1, n] do
    if Eval(D_i) > Eval(D_{i-1}) then
        Generate similar sentences s^b ~ p_{G_i}(·|s^a) from sampled sentences s^a from C;
        Predict pseudo-labels y^i ~ p_{D_i}(·|s^a, s^b);
        Use threshold ε_D to select data on {s^a, s^b, y^i} to train D_{i+1};
        Predict pseudo-labels y^{i+1} ~ p_{D_{i+1}}(·|s^a, s^b);
        Use threshold ε_G and additional rules to select data on {s^a, s^b, y^{i+1}} to train G_{i+1};
    end if
end for
```

4 EXPERIMENTS

4.1 TASKS DESIGN

In the tasks we designed, pretraining dataset is used to warm up the discriminator and generator, and domain-related corpus is a set of independent sentences. To prevent label leakage, none of the train datasets participate in the pre-training of the generator and discriminator. In other words, the datasets in pre-training and self-consistent training are two non-overlapped datasets.

Zero-shot Baseline: We utilize the best-performing Chinese model roberta-wwm-ext-large (Cui et al., 2020; 2021) and English model albert-xxlarge-v2 (Lan et al., 2020) as the base discriminators in our learning framework.

Fine-tune Baseline: We compare our model with several state-of-the-art semantic matching models including: Chinese models macbert (Cui et al., 2020), structbert (Wang et al., 2020), roformer (Su et al., 2021; Su et al., 2022), xlnet (Xu et al., 2020), electra (Cui et al., 2020), albert (Lan et al., 2020), roberta (Cui et al., 2020; 2021) and English models bert (Kenton & Toutanova, 2019), xlm-roberta (Conneau et al., 2020), xlnet (Yang et al., 2019b), electra (Clark et al., 2020), albert (Lan et al., 2020), roberta (Liu et al., 2019). In order to ensure the fairness of the comparison, we use models of almost the same size for comparison.

4.2 EXPERIMENTS SETUP

4.2.1 DATASETS

We conduct experiments on three Chinese semantic matching datasets AFQMC (Xu et al., 2020), CHIP-STS (Zhang et al., 2022a), Chinese-QQP (Wang et al., 2018) and an English semantic matching dataset MRPC (Wang et al., 2018). The more details of the experimental datasets are described in Appendix C.

4.2.2 MODELS

We adopt the well-established Transformer-XL (Dai et al., 2019)/OPT (Zhang et al., 2022b) architecture as the Chinese/English generator.

To enable the generator to generate similar sentences with better linguistic quality, we pre-train a 5 billion Transformer-XL and a 2.7 billion OPT with a mixture of common text and similar sentence pairs. We use the cleaned large-scale Chinese corpus WuDaoCorpora (Yuan et al., 2021) and the cleaned large-scale English corpus WikiText (Merity et al., 2017) as the common text. For the similar sentence pairs, we collect them across various sources and they are formulated into our carefully-designed prompt during pre-training. Our generator uses nucleus sampling (Holtzman et al., 2020) to generate similar sentences during inference phase. More details regarding model pre-training can be found in Appendix B.

4.3 ZERO-SHOT RESULTS

Table 1 shows how the F1 scores of the discriminators vary with the number of self-consistent learning rounds on different datasets in the zero-shot task. Besides, these four datasets represent different fields, which further reflects the robustness of our method in different fields. The specific training settings are recorded in Appendix D.

Table 1: Zero-Shot Performance of the Discriminator. (F1 Score)

Round Num	AFQMC (Financial)	CHIP-STS (Medical)	Chinese-QQP (Common)	MRPC (News)
baseline	38.25	58.82	57.88	68.54
1	39.61	62.89	60.08	75.47
2	44.98	67.24	58.57	76.63
3	45.99	71.38	60.30	83.00
4	45.71	71.45	61.31	83.90
5	48.01	74.06	64.47	84.24
6	50.41	74.08	66.44	84.50
7	50.68	76.66	63.88	84.32
8	**51.36**	76.30	65.46	**84.61**
9	-	76.67	68.08	-
10	-	**77.42**	**70.51**	-
	+13.11%	**+18.60%**	**+12.63%**	**+16.07%**

The last row in Table 1 demonstrates the improvement our self-consistent training made on the last round against the first round. We can see that the F1 score gradually increases after each round of training, and ultimately achieved 10+ absolute percentage(AP) of improvement. We believe what pushes the discriminator to improve are the self-consistent it obtained with generator step by step during the loop.

To verify that the generator improves after self-consistent training, we adopt Perplexity and Bertscore (Zhang et al., 2020) to measure language fluency and semantic similarity(a.k.a domain specificity) respectively. For different generators in different rounds, we first select s^a in similar sentence pairs from the same testing dataset as the original sentences input, and then use the greedy search algorithm to generate similar sentences s^b. The reason for not using other sampling methods is to ensure reproducibility. Given the generated sentences, we introduce an additional GPT2 [4] model to calculate the perplexity of generated similar sentences and use a third-party library [5] to calculate the bertscore between the original and generated similar sentences. The results are shown in Table 2.

Since lower perplexity indicates more fluent sentence, higher bertscore indicates more similar sentence. We can see that the perplexity/bertscore of the last round in Table 2 has decreased/improved compared to the first round. It shows that after self-consistent training, the generator is indeed gradually getting better. The reason why the improvement of the generator is not as obvious as that of the discriminator is that the model size of the generator is several times that of the discriminator, and the number of training samples is not sufficient. In Appendix E we show the generation samples of different rounds of generators to compare the changes of the generators.

[4]Wenzhong-GPT2-110M (Wang et al., 2022) for Chinese data, and gpt2-base (Radford et al., 2019) for English data.

[5]https://pypi.org/project/bert-score/

Table 2: Zero-Shot Performance of the Generator.

	AFQMC	CHIP-STS	Chinese-QQP	MRPC
Perplexity *-first round*	10.13	6.86	12.94	28.71
Perplexity *-last round*	8.43	5.97	12.27	17.56
Bertscore *-first round*	0.79	0.84	0.87	0.94
Bertscore *-last round*	0.80	0.85	0.89	0.97

4.4 FINE-TUNE RESULTS

It is worth mentioning that our method is not only effective in the zero-shot case, but also achieves good results in the fine-tune case. In order to ensure the objectiveness and fairness of the comparison process, we reproduce several Chinese and English state-of-the-art semantic matching models, and their performances on different testing datasets are shown in Table 3.

Table 3: F1 Score of Different Discriminators on the Testing Dataset.

	#Param(zh/en)	AFQMC	CHIP-STS	Chinese-QQP	MRPC	AVG
bert-large	-/335M	-	-	-	82.51	-
xlm-roberta	-/278M	-	-	-	84.27	-
macbert-large	326M/-	61.11	85.94	72.94	-	73.33
structbert-large	326M/-	60.56	85.17	76.33	-	74.02
roformer-large	316M/-	**64.19**	84.16	**76.56**	-	74.97
xlnet-large	360M/360M	50.31	82.97	64.96	79.51	69.44
electra-large	324M/334M	54.59	84.97	71.81	89.64	75.25
albert-xxlarge	221M/223M	56.87	86.32	70.52	**91.21**	76.23
roberta-large	326M/355M	57.29	**86.93**	74.58	90.24	**77.26**
Our Method	-	**66.59**	**88.39**	**78.43**	**92.78**	**81.55**

Our approach uses the best-performing model on each dataset as the base discriminator for self-consistent learning. The last row in Table 3 shows that our method outperforms the state-of-the-art semantic matching models by 1 or 2 AP on all four datasets. In other words, our method achieves state-of-the-art performance on the current semantic matching task.

4.5 EVALUATING SELF-CONSISTENCY

In this section, we will evaluate the consistent between generator and discriminator as the learning loop unfolds. We follow the same method used in Section 4.3 and use greedy search to generate similar sentences on the same test dataset. Then we calculate the score $R_{\mathcal{D}}$ of the discriminator according to Equation 7 for the original sentences s^a and the generated similar sentences s^b.

$$R_{\mathcal{D}} = p_{\mathcal{D}}(y^+|s^a, s^b) \tag{7}$$

where y^+ represents the discriminator predicts a positive example.

However, for the generator, we do not have a reliable method to measure whether s^a and s^b are similar by using the generative model itself. So in order to quantify this similarity, we introduce a static third-party model SimCSE [6] to get the hidden representation $\mathbf{A}, \mathbf{B}$ of the sentences s^a, s^b. The cosine similarity $R_{\mathcal{G}}$ between $\mathbf{A}$ and $\mathbf{B}$ is then calculated according to Equation 9 to approximate the score of the generator.

[6]We use SimCSE-bert-base to evaluate Chinese datasets and sup-simcse-bert-base-uncased to process English ones. (Gao et al., 2021)

$$\mathbf{A}, \mathbf{B} = \text{Encoder}(s^a), \text{Encoder}(s^b) \tag{8}$$

$$R_{\mathcal{G}} = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|_2 * \|\mathbf{B}\|_2} \tag{9}$$

where $\mathbf{A}$ and $\mathbf{B}$ both represent the embedding representation at the $[CLS]$ position. Note that in each round, the original sentence s^a remains unchanged, while the generated sentence s^b changes.

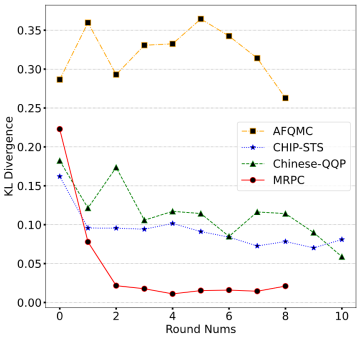

Figure 2: The variation of KL divergence with the number of rounds.

	KL Divergence *-first round*	KL Divergence *-last round*
AFQMC	0.29	0.26
CHIP-STS	0.16	0.08
Chinese-QQP	0.18	0.06
MRPC	0.22	0.02

Table 4: The KL divergence of the first and last rounds.

Therefore, we can get two groups of score distributions, which represent the average similarity score of sentence pairs $\{s^a, s^b\}$ for each round of generator and discriminator. If the distributions are getting closer as the number of training rounds increases, it means that the distributions are getting more consistent. In other words, if the results show that the distance between the two sets of distributions is getting smaller, it demonstrates that the generator and discriminator are indeed gradually showing agreement.

According to the proof in Appendix A, we take KL divergence as evaluation criteria. The variation of the KL divergence with the number of training rounds is shown in Figure 2. All the curves (orange: AFQMC, blue: CHIP-STS, green: Chinese-QQP, red: MRPC) show a clear downward trend, indicating that the distance between the two data distributions decreases with the increase of the number of training rounds. It further shows that the generator and discriminator do gradually show consistent as the training progresses. Table 4 shows the distance values of the two groups of probability distributions in the first and last rounds. It can be observed that the four datasets show a significant decrease in the distance.

4.6 Effect of Adding Pretraining Data and Selection Mechanism

We perform ablation experiments on pretraining data and selection mechanism in the zero-shot case. As described in Section 4.1, the pretraining data is used to pre-train the generator and discriminator, completely independent of the experimental dataset in self-consistent training.

In order to explore the influence of pretraining data during training, we no longer add it in each round when training the discriminator, but only use the generated data. Nevertheless, when the generator is trained, the pretraining data is still retained, to prevent it from learning from self-generated data only every time. The result of removing the pretraining data is shown by the green line in Figure 3.

With all other parameters being the same, the discriminator is slightly less effective after the same number of training rounds compared with the original method (red line in Figure 3). However, the green line maintains a certain upward trend, and converges closely with the red lines in most datasets. This indicates that the pretraining data does not play a key role in improving the discriminator, but rather it is the generated data that takes effect.

In order to explore the influence of selection mechanism during training, we no longer add the selection mechanism when training the discriminator, and the training of the generator is still unchanged. The result of removing the selection mechanism is shown as the blue line in Figure 3.

Compared with the original method of retaining the selection mechanism (red line in Figure 3), it is evident that if the selection mechanism is removed, the effect of the discriminator remains almost unchanged after the first round of training improvement. It verifies that the selection mechanism plays an important role in the self-consistent learning framework. In addition, we output the results of the discriminator without the selection mechanism and find that it labels almost all samples in the testing dataset as positive. In other words, the f1 score on the blue line does not represent the true ability of the discriminator.

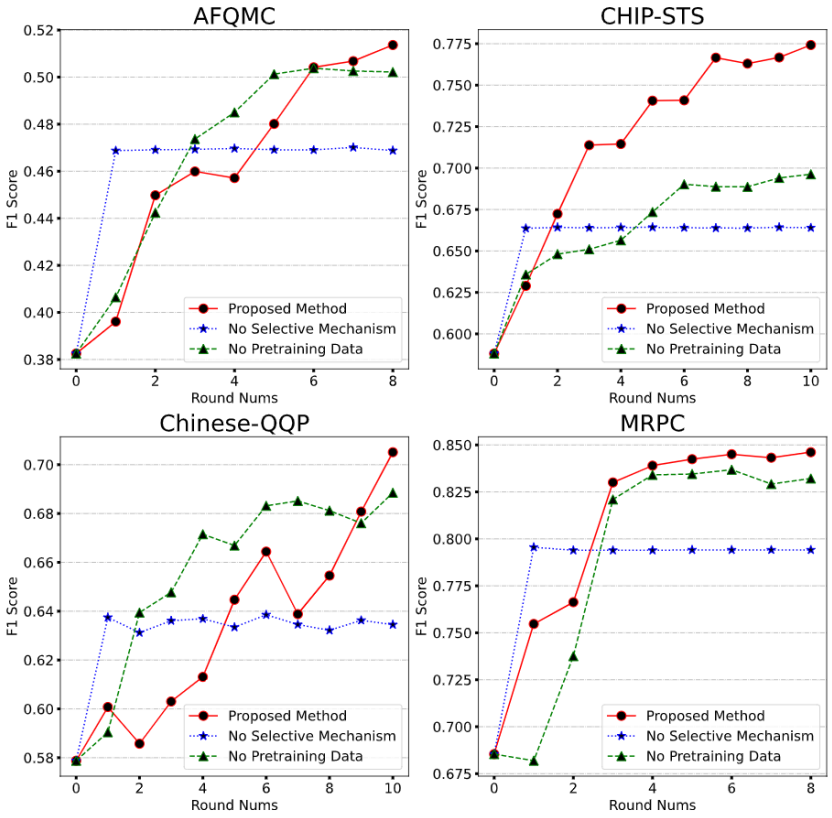

Figure 3: Results of ablation experiments on pretraining data and selection mechanism. The red line represents the result of our proposed method, the blue line represents the result after removing the selection mechanism, and the green line represents the result after removing the pretraining data.

5 Conclusion

In this paper, we propose a semi-supervised self-consistent learning framework which can utilize both limited labeled pretraining data and large-scale unlabeled domain-related corpus to realize similar-sentence generation and semantic-matching discrimination. In the process of training the closed loop, our framework can make the generator and the discriminator mutually enhance each other and reach the optimal simultaneously at the end. Experiments on Chinese and English datasets demonstrate that our proposed framework can indeed improve both generative and discriminative models simultaneously.

For future work, we will explore the effectiveness of applying other NLP tasks to our self-consistent learning framework, which is general and has no requirements on generators and discriminators. In addition, there is still room to improve the performance of the zero-shot task to the level of the fine-tune task.

REFERENCES

David Alvarez-Melis, Vikas Garg, and Adam Tauman Kalai. Why gans are overkill for nlp. *arXiv preprint arXiv:2205.09838*, 2022.

Eric Arazo, Diego Ortego, Paul Albert, Noel E O'Connor, and Kevin McGuinness. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE, 2020.

Martin Arjovsky and Léon Bottou. Towards principled methods for training generative adversarial networks. *arXiv preprint arXiv:1701.04862*, 2017.

Amin Banitalebi-Dehkordi and Yong Zhang. Repaint: Improving the generalization of downstream visual tasks by generating multiple instances of training examples. In *32nd British Machine Vision Conference 2021, BMVC 2021, Online, November 22-25, 2021*, pp. 122. BMVA Press, 2021. URL https://www.bmvc2021-virtualconference.com/assets/papers/0068.pdf.

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. Language models are few-shot learners. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pp. 2633–2650, 2021.

Tong Che, Yanran Li, Ruixiang Zhang, R Devon Hjelm, Wenjie Li, Yangqiu Song, and Yoshua Bengio. Maximum-likelihood augmented discrete generative adversarial networks. *arXiv preprint arXiv:1702.07983*, 2017.

Jing Chen, Qingcai Chen, Xin Liu, Haijun Yang, Daohe Lu, and Buzhou Tang. The BQ corpus: A large-scale domain-specific Chinese corpus for sentence semantic equivalence identification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 4946–4951, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1536. URL https://aclanthology.org/D18-1536.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. Palm: Scaling language modeling with pathways. *CoRR*, abs/2204.02311, 2022. doi: 10.48550/arXiv.2204.02311. URL https://doi.org/10.48550/arXiv.2204.02311.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. ELECTRA: pre-training text encoders as discriminators rather than generators. In *8th International Conference*

on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020. OpenReview.net, 2020. URL https://openreview.net/forum?id=r1xMH1BtvB.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Édouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 8440–8451, 2020.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Shijin Wang, and Guoping Hu. Revisiting pre-trained models for Chinese natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pp. 657–668, Online, November 2020. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.findings-emnlp.58.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, and Ziqing Yang. Pre-training with whole word masking for chinese bert. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3504–3514, 2021. doi: 10.1109/TASLP.2021.3124365.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G Carbonell, Quoc Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 2978–2988, 2019.

Emily L Denton, Soumith Chintala, and Rob Fergus. Deep generative image models using a laplacian pyramid of adversarial networks. *Advances in neural information processing systems*, 28, 2015.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pp. 4171–4186. Association for Computational Linguistics, 2019. doi: 10.18653/v1/n19-1423. URL https://doi.org/10.18653/v1/n19-1423.

Varun Gangal, Steven Y Feng, Malihe Alikhani, Teruko Mitamura, and Eduard Hovy. Nareor: The narrative reordering problem. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 10645–10653, 2022.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 6894–6910, 2021.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014. URL https://arxiv.org/abs/1406.2661.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. *Advances in neural information processing systems*, 30, 2017.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rygGQyrFvH.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparametrization with gumble-softmax. In *International Conference on Learning Representations (ICLR 2017)*. OpenReview. net, 2017.

Jisoo Jeong, Seungeui Lee, Jeesoo Kim, and Nojun Kwak. Consistency-based semi-supervised learning for object detection. *Advances in neural information processing systems*, 32, 2019.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pp. 4171–4186, 2019.

Abhishek Kumar, Prasanna Sattigeri, and Tom Fletcher. Semi-supervised learning with gans: Manifold invariance with improved inference. *Advances in neural information processing systems*, 30, 2017.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=H1eA7AEtvS.

Xin Liu, Qingcai Chen, Chong Deng, Huajun Zeng, Jing Chen, Dongfang Li, and Buzhou Tang. LCQMC:a large-scale Chinese question matching corpus. In *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 1952–1962, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics. URL https://aclanthology.org/C18-1166.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv e-prints*, pp. arXiv–1907, 2019.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=Byj72udxe.

Ehsan Montahaei, Danial Alihosseini, and Mahdieh Soleymani Baghshah. Dgsan: discrete generative self-adversarial network. *Neurocomputing*, 448:364–379, 2021.

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

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67, 2020. URL http://jmlr.org/papers/v21/20-074.html.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016.

Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*, 33:596–608, 2020.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. RoFormer: Enhanced Transformer with Rotary Position Embedding. *arXiv e-prints*, art. arXiv:2104.09864, April 2021.

Jianlin Su, Shengfeng Pan, Bo Wen, and Yunfeng Liu. Roformerv2: A faster and better roformer - zhuiyiai. Technical report, 2022. URL https://github.com/ZhuiyiTechnology/roformer-v2.

Jesper E Van Engelen and Holger H Hoos. A survey on semi-supervised learning. *Machine Learning*, 109(2):373–440, 2020.

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

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 353–355, Brussels, Belgium, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5446. URL https://aclanthology.org/W18-5446.

Junjie Wang, Yuxiang Zhang, Lin Zhang, Ping Yang, Xinyu Gao, Ziwei Wu, Xiaoqun Dong, Junqing He, Jianheng Zhuo, Qi Yang, Yongfeng Huang, Xiayu Li, Yanghan Wu, Junyu Lu, Xinyu Zhu, Weifeng Chen, Ting Han, Kunhao Pan, Rui Wang, Hao Wang, Xiaojun Wu, Zhongshen Zeng, Chongpei Chen, Ruyi Gan, and Jiaxing Zhang. Fengshenbang 1.0: Being the foundation of chinese cognitive intelligence. *arXiv preprint arXiv:2209.02970*, 2022.

Wei Wang, Bin Bi, Ming Yan, Chen Wu, Jiangnan Xia, Zuyi Bao, Liwei Peng, and Luo Si. Structbert: Incorporating language structures into pre-training for deep language understanding. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=BJgQ4lSFPH.

Zirui Wang, Adams Wei Yu, Orhan Firat, and Yuan Cao. Towards zero-label language learning. *arXiv preprint arXiv:2109.09193*, 2021.

Maciej Wiatrak, Stefano V Albrecht, and Andrew Nystrom. Stabilizing generative adversarial networks: A survey. *arXiv preprint arXiv:1910.00927*, 2019.

Qingyang Wu, Lei Li, and Zhou Yu. Textgail: Generative adversarial imitation learning for text generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 14067–14075, 2021.

Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. Unsupervised data augmentation for consistency training. *Advances in Neural Information Processing Systems*, 33:6256–6268, 2020.

Liang Xu, Hai Hu, Xuanwei Zhang, Lu Li, Chenjie Cao, Yudong Li, Yechen Xu, Kai Sun, Dian Yu, Cong Yu, Yin Tian, Qianqian Dong, Weitang Liu, Bo Shi, Yiming Cui, Junyi Li, Jun Zeng, Rongzhao Wang, Weijian Xie, Yanting Li, Yina Patterson, Zuoyu Tian, Yiwen Zhang, He Zhou, Shaoweihua Liu, Zhe Zhao, Qipeng Zhao, Cong Yue, Xinrui Zhang, Zhengliang Yang, Kyle Richardson, and Zhenzhong Lan. CLUE: A Chinese language understanding evaluation benchmark. In *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 4762–4772, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.419. URL https://aclanthology.org/2020.coling-main.419.

Yinfei Yang, Yuan Zhang, Chris Tar, and Jason Baldridge. PAWS-X: A cross-lingual adversarial dataset for paraphrase identification. 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)*, pp. 3687–3692, Hong Kong, China, November 2019a. Association for Computational Linguistics. doi: 10.18653/v1/D19-1382. URL https://aclanthology.org/D19-1382.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32, 2019b.

Haiyan Yin, Dingcheng Li, Xu Li, and Ping Li. Meta-cotgan: A meta cooperative training paradigm for improving adversarial text generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 9466–9473, 2020.

Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. Seqgan: Sequence generative adversarial nets with policy gradient. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

Sha Yuan, Hanyu Zhao, Zhengxiao Du, Ming Ding, Xiao Liu, Yukuo Cen, Xu Zou, Zhilin Yang, and Jie Tang. Wudaocorpora: A super large-scale chinese corpora for pre-training language models. *AI Open*, 2:65–68, 2021. ISSN 2666-6510. doi: https://doi.org/10.1016/j.aiopen.2021.06.001. URL https://www.sciencedirect.com/science/article/pii/S2666651021000152.

Ningyu Zhang, Mosha Chen, Zhen Bi, Xiaozhuan Liang, Lei Li, Xin Shang, Kangping Yin, Chuanqi Tan, Jian Xu, Fei Huang, Luo Si, Yuan Ni, Guotong Xie, Zhifang Sui, Baobao Chang, Hui Zong,

Zheng Yuan, Linfeng Li, Jun Yan, Hongying Zan, Kunli Zhang, Buzhou Tang, and Qingcai Chen. CBLUE: A Chinese biomedical language understanding evaluation benchmark. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7888–7915, Dublin, Ireland, May 2022a. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.544. URL https://aclanthology.org/2022.acl-long.544.

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

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=SkeHuCVFDr.

A PROOFS

A.1 PROOF OF THEOREM 1

Theorem 1. At round t, the aim of the optimization of generator $\mathcal{G}_\theta^t$, given the previous round discriminator $\mathcal{D}_\phi^{t-1}$, boils down to,

$$\min_\theta \mathbb{D}_{KL}(p_{\mathcal{D}_\phi^{t-1}}(\cdot), p_{\mathcal{G}_\theta^t}(\cdot))$$

Proof. The samples generated by the previous round generator $\mathcal{G}_\theta^{t-1}$ and filtered by the previous round discriminator $\mathcal{D}_\phi^{t-1}$ with a threshold ϵ_{t-1} are used for the training of the current round generator $\mathcal{G}_\theta^{t-1}$, the goal of which is to maximize the probability that the generated samples pass the discrimination for the fixed $\mathcal{D}_\phi^{t-1}$,

$$\max_\theta \mathbb{E}_{x\sim p_{\mathcal{G}_\theta^t}} p_{\mathcal{D}_\phi^{t-1}}(x)$$

A further transformation of above formula shows that

$$\begin{aligned}
&\max_\theta \mathbb{E}_{x\sim p_{\mathcal{G}_\theta^t}} p_{\mathcal{D}_\phi^{t-1}}(x) \\
\Rightarrow &\max \int \mathrm{d}\boldsymbol{\theta} \nabla_{\boldsymbol{\theta}} \mathbb{E}_{x\sim p_{\mathcal{G}_\theta^t}} p_{\mathcal{D}_\phi^{t-1}}(x) \\
\Rightarrow &\max_\theta \int \mathrm{d}\boldsymbol{\theta} \mathbb{E}_{x\sim p_{\mathcal{G}_\theta^t}} \nabla_{\boldsymbol{\theta}} \log p_{\mathcal{G}_\theta^t}(x) p_{\mathcal{D}_\phi^{t-1}}(x) \\
\Rightarrow &\max_\theta \int \mathrm{d}\boldsymbol{\theta} \nabla_{\boldsymbol{\theta}} \frac{1}{K} \sum_{k=1}^{K} \log p_{\mathcal{G}_\theta^t}(x_k) p_{\mathcal{D}_\phi^{t-1}}(x_k) - \log p_{\mathcal{D}_\phi^{t-1}}(x_k) p_{\mathcal{D}_\phi^{t-1}}(x_k) \\
\Rightarrow &\min_\theta \mathbb{D}_{KL}(p_{\mathcal{D}_\phi^{t-1}}(\cdot), p_{\mathcal{G}_\theta^t}(\cdot))
\end{aligned}$$

which concludes our proof.

B MODEL TRAINING DETAILS

Our 5B Transformer-XL is pre-trained on 32 A100s with 40G memory for 45 days, the batch size is set to 32*8, and after running 44.5w steps, the final validation loss is reduced to about 2.4 to stop training. And our 2.7B OPT is incrementally trained on the basis of the open source model.

Before training the generator, we ingeniously utilize the mechanism of Transformer-XL's memory attention mask to concatenate the input sentences, so as to reduce the number of padding in a batch and achieve the purpose of saving GPU memory. During training of the generator, we add noise to the original sentence, randomly replaced or discarded some tokens with a certain probability. In addition, the Chinese and English prompt design of Transformer-XL is as follows:

- Chinese prompt: "s^a"的相似句是"s^b" (en: A similar sentence to "s^a" is "s^b".)
- English prompt: "s^a" is similar to "s^b"

When training the discriminator, we follow the separator design in BERT (Devlin et al., 2019), use $[SEP]$ to concatenate the two sentences, and take the embedding at the $[CLS]$ position to represent the sentence embedding of the whole sentence to predict the label. Moreover, we utilize the mask method in BERT to randomly mask 15% of the input tokens.

C DATASET SETTING DETAILS

The statistics of the experimental datasets are reported in Table 5.

We collect other Chinese dataset (LCQMC (Liu et al., 2018), OPPO, PAWS-X-zh (Yang et al., 2019a), BQ (Chen et al., 2018), CCKS, Chinese-STS-B (Wang et al., 2018)) and English dataset

Table 5: Statistics of Experiment Datasets

		AFQMC	CHIP-STS	Chinese-QQP	MRPC(en)
Zero-Shot	domain-related corpus	68668	32000	10800	8,152
	test dataset	4316	4000	3600	1725
Fine-Tune	train dataest	34334	16000	5400	4076
	test dataset	4316	4000	3600	1725

(QQP (Wang et al., 2018), STS-B (Wang et al., 2018), PAWS-X-en (Yang et al., 2019a)) as much as possible, but they are only used as similar sentence corpus for pre-training generator. To avoid label leakage during evaluation, we no longer use these datasets as our final testing dataset.

Our Chinese-QQP is 9k pieces of data randomly translated from English QQP. In order to ensure that the corpus of similar sentences for pre-training the generator is sufficient, we choose two datasets, QQP(149,263) and PAWS-X-en(23,599), which have more data, as pre-training data. At the same time, because STS-B dataset is not for binary classification, it is no longer used for testing. To sum up the above, the QQP, STS-B and PAWS-X-en datasets are used as the pre-training data of the generator, and only the samples with similarity labels greater than or equal to 4 in STS-B are selected for training.

D Parameter Settings

The specific parameters of zero-shot are shown in Table 6. The three thresholds are used to select positive and negative examples for training the discriminator and positive examples for training the generator. The number of sentences is the domain-related corpus generated by each input to the generator, and is estimated based on the total number of samples of domain-related corpus. The model sizes of English and Chinese are different, so the set batch size is different. The learning rate change adopts a cosine annealing strategy.

The specific parameters of fine-tune are shown in Table 7, which is basically the same as that of zero-shot. It is worth mentioning that since the two datasets AFQMC and Chinese-QQP have Out Of Memory when training the discriminator, the batch size is halved.

E Generate Samples

Some generated samples in English are shown in Table 8.

Some generated samples in Chinese are shown in Table 9.

Table 6: Parameter Settings of Zero-Shot.

	AFQMC	CHIP-STS	Chinese-QQP	MRPC
Maximum Threshold *-negative*	0.8	0.9	0.95	0.95
Minimum Threshold *-negative*	0.6	0.7	0.8	0.8
Maximum Threshold *-positive*	0.8	0.9	0.95	0.95
Minimum Threshold *-positive*	0.6	0.7	0.8	0.8
Maximum Threshold *-generator*	0.6	0.9	0.95	0.95
Minimum Threshold *-generator*	0.6	0.7	0.8	0.8
Threshold Increase	0.07	0.1	0.05	0.05
Sentence Num	6000	6000	3000	4000
Learning Rate	2e-5			
Warm Up Steps	40			
Early Stopping Patience	1			
Generator Batch Size *-training*	2(concatenate 30 samples)			24
Generator Batch Size *-predicting*	512			100
Discriminator Batch Size *-training*	64			32
Discriminator Batch Size *-predicting*	384			96

Table 7: Parameter Settings of Fine-Tune.

	AFQMC	CHIP-STS	Chinese-QQP	MRPC
Maximum Threshold *-negative*	0.98	0.98	0.84	0.8
Minimum Threshold *-negative*	0.9	0.7	0.6	0.6
Maximum Threshold *-positive*	0.98	0.98	0.98	0.8
Minimum Threshold *-positive*	0.9	0.7	0.9	0.6
Maximum Threshold *-generator*	0.98	0.98	0.98	0.8
Minimum Threshold *-generator*	0.9	0.7	0.9	0.6
Threshold Increase	0.07	0.07	0.07	0.2
Sentence Num	6000	6000	3000	3000
Learning Rate	5e-6			
Warm Up Steps	40			
Early Stopping Patience	1			
Generator Batch Size *-training*	2(concatenate 30 samples)			24
Generator Batch Size *-predicting*	512			100
Discriminator Batch Size *-training*	32	64	32	32
Discriminator Batch Size *-predicting*	256	384	256	96

Table 8: Generated Samples in English

MRPC(News)
origin: Hong Kong was flat, Australia, Singapore and South Korea lost 0.2-0.4 percent. *first round:* the market is flat, the US is flat, Europe is flat, Japan is flat, ... Malaysia is flat, Philippines is flat, Vietnam is flat, Singapore is *last round:* Hong Kong was unchanged, Australia, Singapore and South Korea were down 0.2-0.4 percent.
origin: Ballmer has been vocal in the past warning that Linux is a threat to Microsoft. *first round:* Microsoft is a threat to Linux. *last round:* Ballmer has been vocal in the past about the threat Linux poses to Microsoft.
origin: Last year, Congress passed similar, though less expensive, buyout legislation for peanut farmers, ending that Depression-era program. *first round:* The government is going to buy out your farm and give you a check for $1,000,000.00. You can't refuse. *last round:* Congress passed a similar bill last year to end a Depression-era program for peanut farmers, ending that program.
origin: Licensing revenue slid 21 percent, however, to $107.6 million. *first round:* The company's net income fell to $1.1 million, or 2 cents per share, in the quarter ended Sept. 30, from $1.3 million, or 3 cents per share, a year earlier. *last round:* Licensing revenue fell 21 percent to $107.6 million in the third quarter.
origin: SCO says the pricing terms for a license will not be announced for weeks. *first round:* we're not going to tell you how much it costs to buy a car, but we're going to tell you how much it costs to rent one. *last round:* The pricing terms for a license for the SCO software will not be announced for several weeks.

Table 9: Generated Samples in Chinese.

AFQMC (Financial)	*origin*：可以用自己的支付宝为别人的花呗还款吗 (Can I use my Alipay to repay someone else's Huabei) *first round*：花呗的钱可以让其他人代还吗 (Can someone else pay for the money from Huabei) *last round*：花呗能不能帮别人还 (Can my Huabei help others repay?)
	origin：花呗怎么还欠费 (How does Huabei pay off the arrears) *first round*：花呗怎么还有电费 (Why does Huabei still have electricity bills) *last round*：花呗还款怎么还 (How to pay off Huabei)
	origin：我的花呗收款二维码 (QR code for my Huabei payment) *first round*：商家花呗的二维码 (Merchant Huabei's QR code) *last round*：花呗收款二维码在哪里 (Where is the QR code for Huabei payment)
CHIP-STS (Medical)	*origin*：艾滋病的病因是什么 (What is the cause of AIDS) *first round*：艾滋病毒是什么？ (What is HIV?) *last round*：艾滋病是什么原因引起的? (What causes AIDS?)
	origin：高血压总是流口水是怎么回事 (High blood pressure is always drooling what is going on) *first round*：高血压怎么回事 (What about high blood pressure) *last round*：高血压为什么会流口水? (Why does high blood pressure cause drooling?)
	origin：得了糖尿病,现在越来越瘦了怎么回事 (Why am I getting thinner and thinner now that I have diabetes) *first round*：糖尿病现在怎么回事? (What's going on with diabetes now?) *last round*：糖尿病患者为什么会瘦? (Why do people with diabetes lose weight?)
Chinese-QQP (Common)	*origin*：如何从此网站删除我的帐户？ (How do I delete my account from this site?) *first round*：怎么删除网站 (How to delete a website) *last round*：如何才能删除我的帐户？ (How can I delete my account?)
	origin：关于电子产品的一些好书是什么？ (What are some good books on electronics?) *first round*：有什么好的电子产品推荐 (Any good electronics recommendations) *last round*：有哪些关于电子产品的好书? (What are some good books about electronics?)
	origin：为什么没有人看到无尽和无限之间的区别? (Why does no one see the difference between endless and infinite?) *first round*：为什么宇宙中没有极限的存在 (Why is there no limit in the universe) *last round*：为什么没有人知道无限和有限之间的区别? (Why does no one know the difference between infinite and finite?)