CoCGAN: Contrastive Learning for Adversarial Category Text Generation

Xin Sheng[1,3], **Linli Xu**[1,3]*, **Yinlong Xu**[2], **Changcun Bao**[4], **Huang Chen**[4], **Bo Ren**[4]
[1] Anhui Province Key Laboratory of Big Data Analysis and Application,
School of Computer Science and Technology, University of Science and Technology of China.
[2] School of Computer Science and Technology, University of Science and Technology of China.
[3] State Key Laboratory of Cognitive Intelligence. [4] Tencent Youtu Lab.
xins@mail.ustc.edu.cn, {linlixu,ylxu}@ustc.edu.cn,
{changcunbao,huaangchen,timren}@tencent.com

Abstract

The task of generating texts of different categories has attracted more and more attention in the area of natural language generation recently. Meanwhile, generative adversarial net (GAN) has demonstrated its effectiveness on text generation, and is further applied to category text generation in later works. Different from existing methods, which mainly consider the pairwise relations between the text embedding and the corresponding fixed one-hot class label (data-to-class relations), this paper proposes a novel Contrastive Category Generative Adversarial Net (CoCGAN) to incorporate contrastive learning into adversarial category text generation, considering more flexible data-to-class relations as well as relations between the multiple text embeddings in the same batch (data-to-data relations). The discriminator of CoCGAN discriminates the authenticity of given samples and optimizes a contrastive learning objective to capture both more flexible data-to-class relations and data-to-data relations among training samples. Accordingly, the generator tries to produce more realistic samples which can confuse the discriminator. Experimental results on both synthetic and real category text generation datasets demonstrate that CoCGAN can achieve significant improvements over the baseline category text generation models.

1 Introduction

Category text generation is the task of generating coherent and meaningful text with different categories and has received increasing attention in many natural language processing applications, such as sentiment analysis (Li et al., 2018) and dialogue generation (Li et al., 2017). It is a further expression of machine intelligence, and makes the generated texts more friendly to humans. Category text generation is a generalization of sentimental text generation and can be seen as a type of conditional text generation. This task focuses on how to integrate the auxiliary category information with the conventional text generation frameworks. Recently, generative adversarial net (GAN) (Goodfellow et al., 2014), where the discriminator is designed to guide the generator, is combined with the reinforcement learning (RL) (Williams, 1992) methods to generate discrete text sequences for general text generation with some remarkable successes (Yu et al., 2017; Guo et al., 2018; Caccia et al., 2020). Different from the general text generation task, category text generation mainly focuses on automatically generating a variety of controllable texts according to the specified categories. However, it is challenging to incorporate the category information into the sentences and design an appropriate objective for generating texts of different categories simultaneously.

In previous works, attempts have been made to extend the general text generation models to category text generation (Wang and Wan, 2018; Liu et al., 2020; Li et al., 2018). However, they only consider relations between the text embeddings and the class labels with simple constraints (data-to-class relations). Among them, SentiGAN (Wang and Wan, 2018) heavily relies on the discriminator based on a $(k+1)$-class classifier, which classifies between "generated" and k real classes. But it ignores the fact that each generated sample involves the degree of authenticity as well as the probability of belonging to a certain category simultaneously, therefore it is less reasonable to directly set the discriminator as a $(k+1)$-class classifier. To address this issue, CSGAN (Li et al., 2018) splits the discriminator into an authenticity discriminator and a category classifier, and optimize the generator with reward-based policy gradient strategy. Nevertheless, both the discriminators of SentiGAN and CSGAN are still limited by the simple category constraints (i.e., cross-entropy loss using fixed one-

*Corresponding author

Proceedings of the 29th International Conference on Computational Linguistics, pages 6403–6414
October 12–17, 2022.

hot class label as target). More recently, CatGAN proposes a category-aware model with a generator based on the relational memory core (RMC) to generate category texts. However, its discriminator still focuses on vanilla data-to-class relations and ignores relations between these text embeddings in the same batch (data-to-data relations).

In this paper, inspired by recent application of contrastive learning in conditional image generation (Kang and Park, 2020), we propose a novel adversarial category text generation framework, namely Contrastive Category Generative Adversarial Net (CoCGAN), to further exploit the category information. Different from SentiGAN, which uses multiple generators, we adopt a conditional generator (Hochreiter and Schmidhuber, 1997) to simplify the model, where an additional class label embedding is set as input to control the type of the generated category text. Following (Yu et al., 2017), we consider the sequence generation procedure as a sequential decision making process and the generator is regarded as a stochastic parametrized policy. Regarding the discriminator, since each generated sample is associated with a real class label and it is rather rough to simply mark it as "generated", the proposed discriminator is divided into two parts: an authenticity discriminator and a contrastive category projector. The authenticity discriminator is a conventional GAN discriminator which is designed as a binary classifier to judge whether the input text is real or not. As for the contrastive category projector, we abandon the conventional way which adopts the cross-entropy loss as the training objective and take class label embeddings into account for more flexible learning. Specifically, the discriminator leverages contrastive learning to pull the text embeddings closer to their corresponding class label embeddings. Furthermore, we also consider relations between the text embeddings which share the same class labels. In other words, the contrastive category projector aims to pull the multiple text embeddings, which are in the same batch, closer to each other when their class labels are the same, while pushing away from each other otherwise. As benefits of the novel contrastive learning paradigm, the discriminator can capture not only more flexible data-to-class relations but also data-to-data relations among training samples. During adversarial training, we adopt Monte Carlo tree search to approximate the state-action value function and the penalty-based (Wang and Wan, 2018) training objective is used to update the generator with policy gradient strategy (Sutton et al., 2000), where we integrate the output of the authenticity discriminator and the contrastive category projector to obtain the overall penalty.

We conduct category text generation experiments on both synthetic and real category datasets and adopt multiple metrics to evaluate the quality of the generated texts. We also compare the proposed CoCGAN with several state-of-the-art category text generation models, including SentiGAN and CatGAN. Experimental results on three datasets (i.e., movie reviews, amazon reviews and synthetic datasets) demonstrate that our model consistently outperforms the state-of-the-art models.

The contributions of this work are three-fold:

- We adopt the conditional generator without loss of generality and decouple the discriminator into two parts, which comprehensively consider the authenticity and category information of the input text.
- We propose the CoCGAN which adopts contrastive learning to leverage not only more flexible data-to-class relations but also data-to-data relations among samples for category text generation. To the best of our knowledge, this work is the first attempt to introduce contrastive learning for category text generation.
- Extensive experiments are conducted on several datasets and the results from multiple perspectives demonstrate the effectiveness of the proposed model.

2 Related Work

Text generation is an important task in natural language processing and has received more and more attention in many fields recently (Sheng et al., 2020; Bahdanau et al., 2014). Traditional text generation models based on recurrent neural network (RNN) (Graves, 2013) generate each token of a sentence conditioned on the previous tokens and the current hidden state. Nevertheless, the training paradigm maximum likelihood estimation (MLE) may suffer from exposure bias (Bengio et al., 2015), which is due to the inherent difference between the training stage and the inference stage of text generation models trained via MLE. Scheduled sampling is proposed by (Bengio et al., 2015) to solve it but soon proved to be inconsistent (Huszár, 2015). To

alleviate this issue, some works adopt generative adversarial net (GAN) (Goodfellow et al., 2014), which has achieved significant successes on computer vision (Radford et al., 2016; Brock et al., 2018). In GAN, the discriminator learns to distinguish whether a given sample is real or not, and the generator learns to confuse the discriminator by generating high quality data. Nevertheless, GAN is designed for differentiable data, which conflicts with the discrete nature of text generation.

To tackle the above non-differentiability problem, reinforcement learning is introduced in SeqGAN (Yu et al., 2017) and LeakGAN (Guo et al., 2018), where the discriminator guides the generator with the reward signal. And this training paradigm is widely adopted (Sheng et al., 2022). Alternatively, MaskGAN (Fedus et al., 2018) uses the actor-critic algorithm to fill in the missing text conditioned on the surrounding context. RankGAN (Lin et al., 2017) replaces the conventional binary classifier with a ranking model as the discriminator. A different direction to address the non-differentiability problem is approximation methods. Specifically, (Zhang et al., 2017) and (Chen et al., 2018) apply an annealed softmax to approximate the argmax operation. While (Gu et al., 2018) and (Nie et al., 2018) propose to use the Gumbel-Softmax relaxation to approximate the categorical distribution.

The aforementioned methods mostly focus on general text generation. For the task of category text generation, CSGAN (Li et al., 2018) proposes a descriptor which consists of a binary discriminator and a classifier that aims to distinguish the categories of the given sentence. However, the adversarial generator optimization of CSGAN still adopts the reward-based training objective, which is the same as SeqGAN and restricts the diversity of the generated sentences. Meanwhile, SentiGAN (Wang and Wan, 2018) introduces multiple generators where each generator focuses on generating samples with a specified sentiment label. In addition, SentiGAN proposes a novel penalty-based training objective to improve the diversity of the generated samples. However, the multiple generators of SentiGAN will increase the complexity of the model with the increase of category numbers. More recently, CatGAN (Liu et al., 2020) introduces a category-aware model for category text generation. Nevertheless, the discriminator of CatGAN still focuses on vanilla data-to-class relations. As can be noticed, in all the methods discussed above, data-to-data relations among the training batch are ignored. Besides these GAN-based methods, some works also make various attempts to improve the conventional conditional generative models (Keskar et al., 2019; Chan et al., 2021; Li et al., 2020), and they are orthogonal to our model which focuses on improving the discriminator to better guide the conditional generator. And these models can replace the conditional generator of our model for further improvements. Thus, in this work, we mainly focus on the GAN-based methods.

Recently, many unsupervised representation learning methods are proposed based on the principle of contrastive learning (Wu et al., 2018; Bachman et al., 2019; He et al., 2020; Henaff, 2020; Chen et al., 2020a,b), and contrastive learning paradigm is firstly adopted for adversarial image generation in (Kang and Park, 2020). Besides, contrastive learning is also applied to conventional conditional text generative models (Lee et al., 2021; Qian et al., 2022). To effectively leverage the class label information, the proposed CoCGAN makes the first attempt to integrate contrastive learning into the discriminator for adversarial category text generation. By exploring both more flexible data-to-class relations and data-to-data relations with contrastive learning, CoCGAN achieves significant improvements over previous works, including the state-of-the-art model CatGAN.

3 Methodology

In this section, we propose CoCGAN by adapting contrastive learning to adversarial category text generation. We begin with introducing the framework of the generator (Sec. 3.1). Then, in order to consider both more flexible data-to-class relations and data-to-data relations, we split the discriminator into an authenticity discriminator and a contrastive category projector (Sec. 3.2) to introduce a label-incorporated contrastive loss. Accordingly, a penalty-based contrastive learning paradigm is designed to optimize the generator during adversarial training (Sec. 3.3). Finally, we propose the Contrastive Category Generative Adversarial Net (CoCGAN) for category text generation (Sec. 3.4).

3.1 Conditional Generator

Following previous works (Yu et al., 2017; Wang and Wan, 2018; Liu et al., 2020), we adopt the generative model based on recurrent neural net-

work (RNN) (Hochreiter and Schmidhuber, 1997). An RNN maps the input embedding representations $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_T$ of the sequence $x_1, x_2, \ldots, x_T$ into a sequence of hidden states $\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_T$ by using the update function g recursively. Different from the SentiGAN which uses k conventional RNNs to generate texts of k classes, we equip a single conventional RNN with an additional label embedding input $\boldsymbol{c}$ to control the category of the generated text, which can reduce the complexity of the generator, and the hidden state $\boldsymbol{s}_t$ is updated as follows:

$$\boldsymbol{s}_t = g(\boldsymbol{s}_{t-1}, \boldsymbol{x}_t, \boldsymbol{c}) \tag{1}$$

where $\boldsymbol{x}_t$ and $\boldsymbol{c}$ are concatenated as the input at timestep t. Then, a softmax output layer maps the hidden states into the output token distribution:

$$p(y_t|x_1, x_2, \ldots, x_t) = \text{softmax}(\boldsymbol{V}\boldsymbol{s}_t + \boldsymbol{b}) \tag{2}$$

where $\boldsymbol{V}$ and $\boldsymbol{b}$ are weight matrix and bias vector respectively. It is worth noting that the generator can be implemented as most of the RNN variants, such as the Long Short-Term Memory (LSTM) cells (Hochreiter and Schmidhuber, 1997), the gated recurrent unit (GRU) (Cho et al., 2014) and the relational memory core (RMC) (Santoro et al., 2018). For easy comparison with the state-of-the-art model CatGAN, which uses RMC-based generator, we adopt both LSTM and RMC as the generators to conduct experiments.

3.2 Label-incorporated Contrastive Discriminator

In SentiGAN, the discriminator is designed as a $(k+1)$-class classifier to distinguish between the real texts with each category (k classes) and the generated texts (1 class). However, it is overlooked that each generated sample is simultaneously associated with the degree of the authenticity and the probability of belonging to a certain category.

3.2.1 Auxiliary Classification Loss

In order to address the problem mentioned above, we make a clear distinction between the authenticity and the real category of the text. Specifically, as shown in Figure 1(b), we divide the $(k+1)$-class SentiGAN discriminator (Figure 1(a)) into two parts: an authenticity discriminator and a category classifier. The authenticity discriminator is designed as a binary classifier to distinguish between the real and generated samples, while the category classifier is a k-class classifier to distinguish the categories of the given sample. We denote the discriminator as D_ϕ where ϕ is the trainable parameter. Besides, we adopt part of the discriminator network (D_{ϕ_1}) before the fully connected layer as the encoder network and use a multi-layer perceptron network parameterized by φ as the body of the category classifier. The training objective of the discriminator is defined as follows:

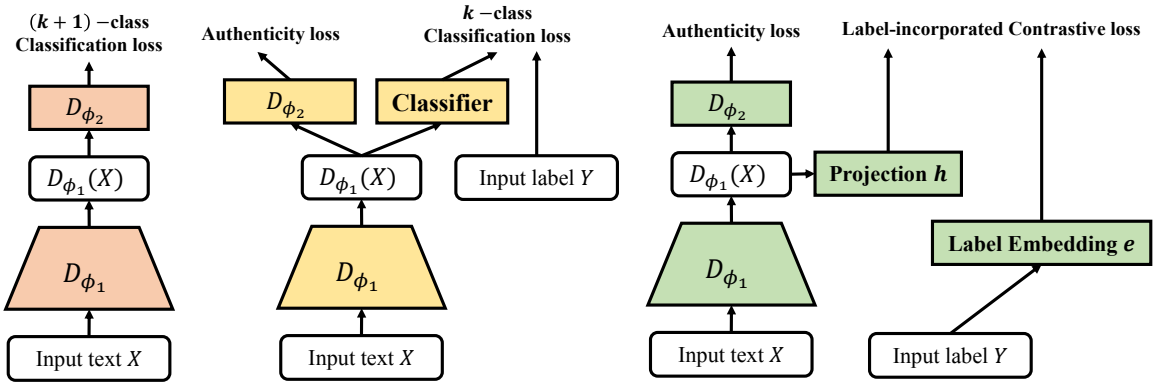
$$\begin{aligned}
\mathcal{L}_{Dis} =& \mathcal{L}_{Auth} + \mathcal{L}_{Cat} \\
\mathcal{L}_{Auth} =& \mathbb{E}_{X\sim P_g} \log D_\phi(X) - \mathbb{E}_{X\sim P_r} \log D_\phi(X) \\
\mathcal{L}_{Cat} =& -\mathbb{E}_{(X,Y)\sim P_r(X,Y)} \log C_{\phi_1,\varphi}(X,Y)
\end{aligned} \tag{3}$$

where $\mathcal{L}_{Auth}$ and $\mathcal{L}_{Cat}$ are the losses of the authenticity discriminator and the category classifier respectively. P_g and P_r represent the generated texts and the real texts respectively, and $P_r(X,Y)$ indicates the real text-label pairs (X and Y represent the real text and the corresponding class label respectively). $D_\phi(X)$ represents the probability that X is real and $C_{\phi_1,\varphi}(X,Y)$ represents the Y-th index of the classifier output, which represents the probability that X belongs to the real Y-th category texts. Here, we comprehensively consider the contributions of the authenticity discriminator and the category classifier. However, the category classifier still uses a cross-entropy loss, which only captures data-to-class relations between a fixed one-hot class label vector and a given text sample.

3.2.2 Label-incorporated Contrastive Loss

As shown in Figure 1(c), to exploit more flexible data-to-class relations and data-to-data relations, we adopt the training paradigm of self-supervised contrastive learning and replace the vanilla k-class classifier with the contrastive category projector.

Firstly, we construct a contrastive learning objective for the discriminator to explicitly control the distances between the text embeddings and the class label embeddings. Different from the conventional contrastive learning NT-Xent loss, which needs appropriate data augmentation and does not take data-to-class relations into account, we leverage the class label embeddings of the categories instead of data augmentation. Given a batch of training text samples $\{X_1, X_2, \ldots, X_m\}$ and the corresponding class labels $\{Y_1, Y_2, \ldots, Y_m\}$, we introduce an encoder $S(\cdot)$ and a projection layer h to map the input text samples to the hypersphere: $l = h(S(\cdot))$. Together with the label embeddings,



(a) SentiGAN (b) CLSGAN (Ours) (c) CoCGAN (Ours)

Figure 1: Schematics of discriminators of three category text GANs. (a) SentiGAN (Wang and Wan, 2018) takes a $(k+1)$-class classifier as its discriminator to guide the generator to generate category texts. (b) CLSGAN improves SentiGAN by explicitly divide the discriminator as a binary authenticity discriminator and a category classifier. (c) The proposed CoCGAN extends the CLSGAN with a label-incorporated contrastive loss, which considers both more flexible data-to-class relations and data-to-data relations in the same batch.

the contrastive loss is defined as follows:

$$\begin{aligned}\ell(X_i, Y_i; \tau) &= -\log \frac{R_{D2C}}{R_{D2C} + R_{D2A}} \\ R_{D2C} &= \exp(l(X_i)^\top e(Y_i)/\tau) \\ R_{D2A} &= \sum_{s=1}^{m} 1_{s \neq i} \cdot \exp(l(X_i)^\top l(X_s)/\tau)\end{aligned} \tag{4}$$

where $e(\cdot)$ denotes the class label embedding function, τ is a temperature scalar to control the pull and push force. (4) pulls the sample X_i nearer to its corresponding class label embedding $e(Y_i)$ and pushes the other samples away. In this work, we adopt part of the discriminator (D_{ϕ_1}) as the encoder $S(\cdot)$ and the multi-layer perceptron network parameterized by φ as the projection layer h to construct the mapping as $l = h(D_{\phi_1}(\cdot))$. To further exploit data-to-data relations, we should also avoid pushing other samples which have the same class label Y_i. Thus, we add cosine similarities of such samples to the numerator in (4) and get the label-incorporated contrastive loss as follows:

$$\begin{aligned}\ell_{\phi_1,\varphi}(X_i, Y_i; \tau) &= -\log \frac{R_{D2C} + R_{D2D}}{R_{D2C} + R_{D2A}} \\ R_{D2D} &= \sum_{s=1}^{m} 1_{Y_s = Y_i, s \neq i} \cdot \exp(l(X_i)^\top l(X_s)/\tau)\end{aligned} \tag{5}$$

where R_{D2C} and R_{D2A} are the same as in (4). Besides reducing the distances between the text embeddings and the corresponding class label embeddings, minimizing (5) will also reduce the distances between the multiple text embeddings with the same class labels while maximizing the others. It is obvious that (5) comprehensively considers more flexible data-to-class relations $l(X_i)^\top e(Y_i)$ and data-to-data relations $l(X_i)^\top l(X_s)$. And the objective of the discriminator can be redefined as:

$$\begin{aligned}\mathcal{L}_{Dis} &= \mathcal{L}_{Auth} + \mathcal{L}_{Cat} \\ \mathcal{L}_{Auth} &= \mathbb{E}_{X \sim P_g} \log D_\phi(X) - \mathbb{E}_{X \sim P_r} \log D_\phi(X) \\ \mathcal{L}_{Cat} &= \mathbb{E}_{(X,Y) \sim P_r(X,Y)} \ell_{\phi_1,\varphi}(X, Y; \tau)\end{aligned} \tag{6}$$

where $P_r(X, Y)$ indicates the real text-label pairs (X and Y represent the real text and the corresponding class label respectively).

3.3 Penalty-based Contrastive Generator Training

For adversarial generator training, instead of the reward-based policy gradient strategy (Yu et al., 2017), we adopt the penalty-based one (Wang and Wan, 2018) to guarantee the diversity of generated text samples. Specifically, the goal of the generator $G_\theta(X|S, Y)$ is to minimize the penalty:

$$\mathcal{L}_{Gen} = \sum_{t=0}^{T-1} G_\theta(X_{t+1}|S_t, Y) \cdot V_{D_{\phi,\varphi}}^{G_\theta}(S_t, Y, X_{t+1}) \tag{7}$$

where T is the length of X, $G_\theta(X_{t+1}|S_t, Y)$ indicates the probability of selecting the $(t+1)$-th word given its current state and class label, denoted as S_t and Y respectively, and $V_{D_\phi}^{G_\theta}(S_t, Y, X_{t+1})$ is

the penalty for the sequence $X_{1:t+1}$, which is calculated by the discriminator. Here Y is the class label corresponding to the label embedding $\boldsymbol{c}$ in (1). Monte Carlo tree search is applied with the roll-out policy G_θ to calculate the penalty function of the generator:

$$V_{D_{\phi,\varphi}}^{G_\theta}(S_{t-1}, Y, X_t) = \begin{cases} \frac{1}{N}\sum_{n=1}^{N}(1-r^n) & t < T \\ 1-r & t = T \end{cases} \tag{8}$$

where T is the length of X, r^n and r are given by the specified discriminator with the n-th Monte Carlo tree search result $X_{1:t}^n$ and the completely generated sentence X as input, respectively.

For the discriminator with auxiliary classification loss proposed in Sec. 3.2.1, r (analogy to r^n) is defined as follows:

$$r = \frac{1}{2}(D_\phi(X) + C_{\phi_1,\varphi}(X, Y)). \tag{9}$$

And for the discriminator with label-incorporated contrastive loss introduced in Sec. 3.2.2, r (analogy to r^n) is redefined as follows:

$$r = \frac{1}{2}(D_\phi(X) + \exp(-\ell_{\phi_1,\varphi}(X, Y; \tau))). \tag{10}$$

It is worth noting that, for both (9) and (10), X represents the generated text and Y is the corresponding input class label.

3.4 Contrastive Category Generative Adversarial Net

With the proposed label-incorporated contrastive loss, we build the framework of CoCGAN. Similar to the training procedure of GAN, CoCGAN has a discriminator step and a generator step which constitute adversarial training. Besides, CoCGAN calculates the label-incorporated contrastive loss with a set of real or generated samples. Algorithm 1 in Appendix A.3 summarizes the complete training procedure of the proposed CoCGAN. We also define the framework with auxiliary classification loss as CLSGAN, and Algorithm 2 in Appendix A.3 shows its training procedure. Different from CSGAN, CLSGAN adopts the penalty-based policy gradient strategy instead of the reward-based one. For both CoCGAN and CLSGAN, the training samples fed into the contrastive category projector and the category classifier are different for discriminator and generator step (i.e., real samples for discriminator step and generated samples for generator step). And for each discriminator iteration, the amount of generated samples is set the same as that of the real samples in a batch to guarantee sufficient training.

In CoCGAN, the discriminator can minimize the distances between the multiple real text embeddings from the same class label while maximizing it otherwise and capture more flexible relations between the real text embeddings and the corresponding class label embeddings. Besides, the relations between the current text embeddings and the wrong class label embeddings are also considered in (5). Specifically, since the wrong class label embeddings are pulled near to their corresponding text embeddings and the text embeddings from wrong class labels are pushed away from the current text embeddings, the wrong class label embeddings are pushed away from the current text embeddings implicitly. Therefore, the discriminator can learn better representations of the given samples, and the conditional generator can be further improved to generate more realistic category texts with the knowledge of the discriminator.

4 Experiments

4.1 Datasets

Without loss of generality, we set the number of categories as 2 and conduct experiments on both synthetic and real datasets, as in previous work (Liu et al., 2020). The synthetic data includes 20,000 samples, and each 10,000 samples are obtained from different oracle-LSTMs to construct category text data. The real data includes two English review datasets: movie reviews (MR) (Socher et al., 2013) and amazon reviews (AR) (McAuley et al., 2015). MR has two sentiment classes (negative and positive), and AR has two types of product reviews (book and application). We follow the same preprocessing steps as in LeakGAN (Guo et al., 2018). MR has 4,503 samples, including 3,153 samples for training and 1,350 samples for testing. As for AR, each review category includes 100,000 samples for training and 10,000 samples for testing, and each sample may consist of multiple sentences.

4.2 Evaluation Metrics

There exist many evaluation metrics to measure the performance of adversarial text generation models. Among them, (Yu et al., 2017) introduces the

Model	SentiGAN	CSGAN	CatGAN	CLSGAN_L	CoCGAN_L	CLSGAN_R	CoCGAN_R
20	6.953	8.522	6.631	6.903	6.611	6.712	**6.314**
40	6.877	7.703	6.392	6.663	6.384	6.445	**6.094**

Table 1: The NLL_oracle scores on synthetic dataset. For the NLL_oracle scores, the lower the better.

Model	MR						AR					
	B@2	B@3	B@4	B@5	N_g	N_d	B@2	B@3	B@4	B@5	N_g	N_d
SentiGAN	0.525	0.287	0.162	0.144	2.501	0.472	0.858	0.811	0.712	0.537	3.367	0.916
CSGAN	0.447	0.199	0.120	0.089	2.937	0.243	0.863	0.677	0.431	0.239	3.373	1.104
CatGAN	0.592	0.330	0.195	0.162	1.679	0.521	0.965	0.910	0.855	0.721	3.143	1.472
CLSGAN_L	0.557	0.327	0.183	0.161	2.313	0.491	0.933	0.892	0.810	0.622	3.306	1.112
CoCGAN_L	0.588	0.342	0.213	0.173	1.686	0.526	0.958	0.913	0.851	0.729	3.152	1.231
CLSGAN_R	0.573	0.344	0.201	0.167	1.958	0.517	0.943	0.903	0.841	0.663	3.195	1.289
CoCGAN_R	**0.632**	**0.383**	**0.227**	**0.182**	**1.462**	**0.536**	**0.984**	**0.957**	**0.882**	**0.764**	**3.024**	**1.537**

Table 2: The comparison of performance on MR and AR. B@n denotes BLEU scores of n-gram. For all BLEU scores, the higher the better. For NLL_gen scores (denoted as N_g), the lower the better. For NLL_div scores (denoted as N_d), the higher the better.

negative log-likelihood NLL_oracle to measure the quality on the synthetic data as follows:

$$\text{NLL}_\text{oracle} = -\mathbb{E}_{Y_\theta \sim P_\theta}[\log P_r(Y_\theta)] \quad (11)$$

where P_θ is the generated data distribution and P_r is the real data distribution.

As for the real data, we adopt NLL_gen (Zhu et al., 2018) and NLL_div (Liu et al., 2020) as the diversity metrics, and define them as follows:

$$\text{NLL}_\text{gen} = -\mathbb{E}_{Y_r \sim P_r}[\log P_\theta(Y_r)], \quad (12)$$
$$\text{NLL}_\text{div} = -\mathbb{E}_{Y_\theta \sim P_\theta}[\log P_\theta(Y_\theta)]. \quad (13)$$

To measure the quality on the real data, since we cannot access the distribution of the real data, we adopt BLEU scores (Zhu et al., 2018) to measure the performance of the models instead of NLL_oracle. And we follow (Liu et al., 2020) to use harmonic mean values of all automatic metrics on each category to evaluate the category text generation models.

4.3 Baselines

We conduct experiments to compare the proposed model with several state-of-the-art methods. For automatic evaluation metrics, we select SentiGAN (Wang and Wan, 2018), CSGAN (Li et al., 2018) and CatGAN (Liu et al., 2020) as baseline models. All models are pre-trained with standard MLE training before adversarial training. All the models are run with 5 random seeds on all experiments and the mean is presented as the final score (see Appendix A.1 for more detailed settings). For the proposed CoCGAN and CLSGAN, we adopt both LSTM and RMC as the generators to conduct experiments.

4.4 Quantitative Results

In this section, for CLSGAN and CoCGAN, we report the results of LSTM-based generator and RMC-based generator (i.e., CLSGAN_L, CLSGAN_R, CoCGAN_L and CoCGAN_R).

4.4.1 Results on the Synthetic Data

We conduct experiments on the synthetic data with the sequence length set as 20 and 40 respectively. Table 1 shows that, CoCGAN equipped with RMC-based generator consistently outperforms other models in terms of NLL_oracle, including the state-of-the-art model CatGAN, which demonstrates that CoCGAN can further exploit the category information with better quality on all categories.

4.4.2 Results on the Real Data

As for the real data (i.e., MR and AR), we use several metrics to measure the quality and the diversity of the generated sentences. After the same preprocessing steps, MR dataset consists of 6,216 unique words with the maximum sentence length 15, and AR dataset contains 6,416 unique words with the maximum sentence length 40. We report the results of CLSGAN and CoCGAN with different generators as on the synthetic data, and the results are presented in Table 2. It is obvious that CLSGAN_L shows its superiority on all metrics compared with CSGAN, since the penalty-based training paradigm adopted by CLSGAN_L can improve the performance compared with the reward-based one of CSGAN. CoCGAN further exploits more flexible data-to-class relations and data-to-data relations to achieve significant improvements over CLSGAN. When equipped with the RMC-

Model	(A)	(B)	(C)	(D)
B@2	0.943	**0.984**	0.952	0.963
B@3	0.903	**0.957**	0.917	0.931
B@4	0.841	**0.882**	0.855	0.847
B@5	0.663	**0.764**	0.734	0.751
N_g	3.195	**3.024**	3.132	3.103
N_d	1.289	**1.537**	1.368	1.475

Table 3: The ablation Study on AR. (A), (B), (C) and (D) represent CLSGAN, CoCGAN, CoCGAN w/o D2D and CoCGAN w/o D2C, respectively. All the models are equipped with RMC-based generator. B@n denotes BLEU scores of n-gram. For all BLEU scores, the higher the better. For NLL_{gen} scores (denoted as N_g), the lower the better. For NLL_{div} scores (denoted as N_d), the higher the better.

τ	0.1	1.0	2.0	4.0
B@2	0.873	**0.984**	0.972	0.951
B@3	0.792	**0.957**	0.938	0.893
B@4	0.698	**0.882**	0.877	0.845
B@5	0.573	**0.764**	0.752	0.710
N_g	3.405	**3.024**	3.121	3.303
N_d	1.372	**1.537**	1.475	1.402

Table 4: Tuning of temperature τ. The experiment is conducted on AR. B@n denotes BLEU scores of n-gram. For all BLEU scores, the higher the better. For NLL_{gen} scores (denoted as N_g), the lower the better. For NLL_{div} scores (denoted as N_d), the higher the better.

based generator, for both CoCGAN and CLSGAN, the performance is further improved compared with the one when equipped with the LSTM-based generator. It is noteworthy that CoCGAN using RMC-based generator outperforms the state-of-the-art model CatGAN with better quality and diversity on all categories as well.

4.4.3 Ablation Study

To investigate the contributions of different parts in the label-incorporated contrastive loss (discussed in Sec. 3.2.2), we conduct ablation study on AR. Here, we remove data-to-class relations from CoCGAN as CoCGAN w/o D2C (i.e., remove R_{D2C} from both denominator and numerator of (5) in Sec. 3.2.2), and data-to-data relations are removed from CoCGAN as CoCGAN w/o D2D (i.e., (4) in Sec. 3.2.2). We also report the results of CLSGAN to compare the performance between the fixed one-hot class label vector and the trainable class label embedding. All the results are shown in Table 3. On the one hand, compared with the fixed one-hot class label vector (i.e., CLSGAN), flexible class label embedding (i.e., CoCGAN w/o D2D) can achieve better performance. On the other hand, data-to-class relations and data-to-data relations are complementary to each other. Without either of them, the performance of CoCGAN shows a significant degradation on all metrics (i.e., both CoCGAN w/o D2D and CoCGAN w/o D2C show worse performance compared with complete CoCGAN). Besides, both CoCGAN w/o D2D and CoCGAN w/o D2C still outperform CLSGAN.

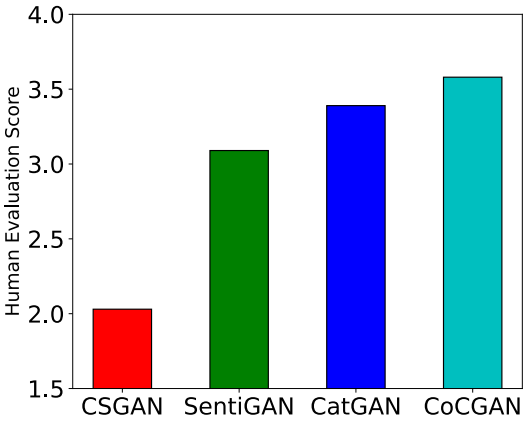


Figure 2: Comparison of human evaluation on a random subset of the AR dataset.

4.4.4 Tuning of Temperature τ

Temperature τ used in (5) is the hyper-parameter to control the pull and push force in contrastive learning and an appropriate temperature can help to capture better data-to-class relations and data-to-data relations. We investigate the impact of τ on AR with a grid search to find a proper value of τ. As shown in Table 4, we experimentally find that the temperature τ of 1.0 yields the best results.

4.5 Qualitative Results

For the qualitative experiments, we adopt RMC-based generator to construct CoCGAN, and only report the results of CoCGAN to compare with baselines (i.e., CSGAN, SentiGAN and CatGAN).

4.5.1 Human Evaluation

To further evaluate the quality of the generated sentences, we randomly select 50 generated samples from each category for human evaluation. The scores from 1 to 5 are assigned to each generated sample, which measures the fluency and the semantic correctness (see Appendix A.2 for more detailed evaluation protocols). The scores of 1 and 5 indicate the worst quality and the best quality

Dataset	SentiGAN	CSGAN	CatGAN	CoCGAN
MR	**Negative:** it's an extremely unpleasant film. **Positive:** it's not smart. **(wrong category)**	**Negative:** an enjoyable experience. **(wrong category)** **Positive:** just intelligence. **(short)**	**Negative:** the movie doesn't add anything fresh to sustain its clever concept. **Positive:** it's a romantic. **(short)**	**Negative:** bad movie is that is, it's just a better travelogue than finding solutions. **Positive:** a very well-made and entertaining picture.
AR	**Book:** i love this book so much. it is one of book that you can not put down. very well written. **Application:** this game is addictive and fun. **(short)**	**Book:** my 4 is on the front of that day seeing travel. just not great, though .. **(Unreadable)** **Application:** this is a fun game. my husband and i both love to play it a lot.	**Book:** this was an awesome book. i loved the book, every page kept me entertained and finished it in two days! **Application:** i really love these games. **(short)**	**Book:** i absolutely loved this book. i am so glad to read the other books in this series. i can't wait for the next one. **Application:** i love this game. it is a great way to pass the time.

Table 5: Generated samples of different models on the real dataset.

respectively. Each generated sample is rated by 10 invited human evaluators who are capable of reading English proficiently. And the harmonic mean values of the average score on each category are shown in Figure 2. It can be observed that, multiple generators help SentiGAN to obtain competitive results, while CatGAN has achieved better performance. And the results of CoCGAN demonstrate that the contrastive learning paradigm helps to consistently outperform these baselines.

4.5.2 Case Study

We select SentiGAN, CSGAN and CatGAN as references to analyze the effectiveness of the proposed label-incorporated contrastive objective. With trained on MR and AR, the generated samples of these models are listed in Table 5, and we can find some problems with the generated sentences of these baselines (e.g., unreadable and wrong category). In contrast, the proposed CoCGAN can produce sentences with better quality.

5 Conclusion

This paper proposes a novel contrastive learning paradigm for adversarial category text generation (CoCGAN). In CoCGAN, a novel label-incorporated contrastive loss is introduced to further exploit more flexible data-to-class relations and data-to-data relations in the training batch, and the category text generation model is enhanced accordingly. It is worth noting that CoCGAN focuses on the perspective of adversarial learning, therefore it is orthogonal to some works which try to optimize the conditonal text generative models themselves, and can be applied to them for further improvements. Extensive experiments demonstrate that our proposed model outperforms the state-of-the-art adversarial category text generation models with better quality and diversity.

Acknowledgements

This research was supported by Anhui Provincial Natural Science Foundation (2008085J31).

References

Philip Bachman, R Devon Hjelm, and William Buchwalter. 2019. Learning representations by maximizing mutual information across views. *Advances in neural information processing systems*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *International Conference on Learning Representations*.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. *Advances in neural information processing systems*.

Andrew Brock, Jeff Donahue, and Karen Simonyan. 2018. Large scale gan training for high fidelity natural image synthesis. *International Conference on Learning Representations*.

Massimo Caccia, Lucas Caccia, William Fedus, Hugo Larochelle, Joelle Pineau, and Laurent Charlin. 2020. Language gans falling short. *International Conference on Learning Representations*.

Alvin Chan, Yew-Soon Ong, Bill Pung, Aston Zhang, and Jie Fu. 2021. Cocon: A self-supervised approach for controlled text generation. *International Conference on Learning Representations*.

Liqun Chen, Shuyang Dai, Chenyang Tao, Dinghan Shen, Zhe Gan, Haichao Zhang, Yizhe Zhang, and Lawrence Carin. 2018. Adversarial text generation

via feature-mover's distance. *Advances in neural information processing systems*.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020a. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, pages 1597–1607. PMLR.

Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. 2020b. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

William Fedus, Ian Goodfellow, and Andrew M Dai. 2018. Maskgan: better text generation via filling in the_. *International Conference on Learning Representations*.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. *Advances in neural information processing systems*, 27.

Alex Graves. 2013. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*.

Jiatao Gu, Daniel Jiwoong Im, and Victor OK Li. 2018. Neural machine translation with gumbel-greedy decoding. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Jiaxian Guo, Sidi Lu, Han Cai, Weinan Zhang, Yong Yu, and Jun Wang. 2018. Long text generation via adversarial training with leaked information. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738.

Olivier Henaff. 2020. Data-efficient image recognition with contrastive predictive coding. In *International Conference on Machine Learning*, pages 4182–4192. PMLR.

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

Ferenc Huszár. 2015. How (not) to train your generative model: Scheduled sampling, likelihood, adversary? *Computing Research Repository*.

Minguk Kang and Jaesik Park. 2020. Contragan: Contrastive learning for conditional image generation. *Advances in neural information processing systems*.

Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. 2019. Ctrl: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858*.

Seanie Lee, Dong Bok Lee, and Sung Ju Hwang. 2021. Contrastive learning with adversarial perturbations for conditional text generation. *International Conference on Learning Representations*.

Jiwei Li, Will Monroe, Tianlin Shi, Sébastien Jean, Alan Ritter, and Dan Jurafsky. 2017. Adversarial learning for neural dialogue generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2157–2169, Copenhagen, Denmark. Association for Computational Linguistics.

Piji Li, Haisong Zhang, Xiaojiang Liu, and Shuming Shi. 2020. Rigid formats controlled text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 742–751, Online. Association for Computational Linguistics.

Yang Li, Quan Pan, Suhang Wang, Tao Yang, and Erik Cambria. 2018. A generative model for category text generation. *Information Sciences*, 450:301–315.

Kevin Lin, Dianqi Li, Xiaodong He, Zhengyou Zhang, and Ming-Ting Sun. 2017. Adversarial ranking for language generation. *Advances in neural information processing systems*.

Zhiyue Liu, Jiahai Wang, and Zhiwei Liang. 2020. Catgan: Category-aware generative adversarial networks with hierarchical evolutionary learning for category text generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8425–8432.

Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, pages 43–52.

Weili Nie, Nina Narodytska, and Ankit Patel. 2018. Relgan: Relational generative adversarial networks for text generation. In *International Conference on Learning Representations*.

Jing Qian, Li Dong, Yelong Shen, Furu Wei, and Weizhu Chen. 2022. Controllable natural language generation with contrastive prefixes. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2912–2924, Dublin, Ireland. Association for Computational Linguistics.

Alec Radford, Luke Metz, and Soumith Chintala. 2016. Unsupervised representation learning with deep convolutional generative adversarial networks. *International Conference on Learning Representations*.

Adam Santoro, Ryan Faulkner, David Raposo, Jack Rae, Mike Chrzanowski, Theophane Weber, Daan Wierstra, Oriol Vinyals, Razvan Pascanu, and Timothy Lillicrap. 2018. Relational recurrent neural networks. *Advances in neural information processing systems*, 31.

Xin Sheng, Linli Xu, Junliang Guo, Jingchang Liu, Ruoyu Zhao, and Yinlong Xu. 2020. Introvnmt: An introspective model for variational neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8830–8837.

Xin Sheng, Linli Xu, Yinlong Xu, Deqiang Jiang, and Bo Ren. 2022. Semantic-preserving abstractive text summarization with Siamese generative adversarial net. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2121–2132, Seattle, United States. Association for Computational Linguistics.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. 2000. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pages 1057–1063.

Ke Wang and Xiaojun Wan. 2018. Sentigan: Generating sentimental texts via mixture adversarial networks. In *IJCAI*, pages 4446–4452.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256.

Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. 2018. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3733–3742.

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

Yizhe Zhang, Zhe Gan, Kai Fan, Zhi Chen, Ricardo Henao, Dinghan Shen, and Lawrence Carin. 2017. Adversarial feature matching for text generation. In *International Conference on Machine Learning*, pages 4006–4015. PMLR.

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

A Appendix

A.1 Experimental Settings

- We implement the baselines based on TextGAN benchmark [1].
- In the CNN based discriminator, the sizes of filter windows are set to be $\{2, 3, 4, 5\}$ and the dimension of each feature map is set as 300.
- The batch size is set to be 64 for all models. The embedding size for the generator is set as 32 and that for the discriminator as 64.
- Adam optimizer with the same hyper-parameters (i.e., $\beta_1 = 0.9$ and $\beta_2 = 0.999$) are employed to optimize all models.
- For MLE pre-training, we run 150 epochs with learning rate set as 0.01.
- For discriminator pre-training, the learning rate is set to be 0.0001.
- For adversarial training, the learning rate is set to be 0.0001 for both the generator and the discriminator.
- All models are trained on a RTX 3090 GPU.

A.2 Human Evaluation Protocols

For category text generation, we conduct human evaluation based on fluency and semantic correctness. The detailed protocols are shown as follows:

- 5-Excellent. Right category, well fluency, and making sense.
- 4-Good. Right category, acceptable fluency with some grammatical errors, and making sense.
- 3-Fair. Right category, no fluency, but conveying some meanings from some parts.
- 2-Poor. Right category, making no sense.
- 1-Bad. Wrong category, making no sense.

[1] https://github.com/williamSYSU/TextGAN-PyTorch

A.3 Algorithms

The training procedures of CoCGAN and CLSGAN are shown in Algorithm 1 and Algorithm 2, respectively. For both CoCGAN and CLSGAN, the training samples fed into the contrastive category projector and the category classifier are different for discriminator step and generator step (i.e., real samples for discriminator step and generated samples for generator step). For sufficient training of discriminator, at each discriminator iteration, half of a training batch are real samples while the other half are generated samples.

Algorithm 1 The training procedure of CoCGAN

Require: Real text dataset T with corresponding class labels; The number of class labels k; Generator G_θ; Discriminator $D_{\phi,\varphi}$; Temperature τ

Initialize G_θ, $D_{\phi,\varphi}$ with random weights
Pre-train G_θ using MLE on T
Generate fake samples F with random class labels using G_θ
Pre-train $D_{\phi,\varphi}$ via minimizing (6) on $\{T, F\}$
while G_θ not converged **do**
  for g-steps **do**
    Generate fake samples F with random class labels using G_θ
    Calculate penalty $V^{G_\theta}_{D_{\phi,\varphi}}$ by (8) and (10) on F
    Update G_θ by minimizing (7)
  end for
  for d-steps **do**
    Generate fake samples F with random class labels using G_θ
    Update $D_{\phi,\varphi}$ via minimizing (6) on $\{T, F\}$
  end for
end while

Algorithm 2 The training procedure of CLSGAN

Require: Real text dataset T with corresponding class labels; The number of class labels k; Generator G_θ; Discriminator $D_{\phi,\varphi}$

Initialize G_θ, $D_{\phi,\varphi}$ with random weights
Pre-train G_θ using MLE on T
Generate fake samples F with random class labels using G_θ
Pre-train $D_{\phi,\varphi}$ via minimizing (3) on $\{T, F\}$
while G_θ not converged **do**
  for g-steps **do**
    Generate fake samples F with random class labels using G_θ
    Calculate penalty $V^{G_\theta}_{D_{\phi,\varphi}}$ by (8) and (9) on F
    Update G_θ by minimizing (7)
  end for
  for d-steps **do**
    Generate fake samples F with random class labels using G_θ
    Update $D_{\phi,\varphi}$ via minimizing (3) on $\{T, F\}$
  end for
end while