Deep Generative Models for Natural Language Generation

Dianqi Li

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2020

Reading Committee:

Ming-Ting Sun, Chair

Xiaodong He

Yejin Choi

Linda Shapiro

Program Authorized to Offer Degree:
Electrical Engineering

University of Washington

Abstract

Deep Generative Models for Natural Language Generation

Dianqi Li

Chair of the Supervisory Committee:
Professor Ming-Ting Sun
Electrical & Computer Engineering

Natural language generation plays an important role in language intelligence, which is an essential topic of artificial intelligence over the past years. Recent advances in generative models combining with deep neural networks have achieved tremendous successes in many natural language generation tasks. Establishing suitable and effective generative models is the key challenge for researchers to fulfill different language generation purposes under varied application scenarios. This thesis focuses on investigating and providing better deep generative models with respect to various natural language generation tasks.

This thesis consists of two parts. The first part explores the ranking-based generative adversarial network for generating texts. We first examine limitations of the commonly used Generative Adversarial Networks (GANs) on text generation tasks, and propose a novel ranking-based generative adversarial network, RankGAN, for generating high-quality language descriptions. Rather than training the discriminator to learn and assign an absolute binary predicate for an individual data sample, the proposed RankGAN is able to analyze and rank a collection of human-written and machine-written sentences by giving a reference group. Concretely, by viewing a set of data samples collectively and evaluating their quality through relative ranking scores, the discriminator is able to make a better assessment which in turn helps to learn a better generator for text generation tasks. We then take a step further to apply RankGAN in image captioning. We explore how to generate captions that are not

only accurate in describing an image but also diverse across different images. By ranking human-written captions above image-mismatched captions within the image-caption joint space, the corresponding caption generator effectively exploits the inherent characteristics of human languages, and generates more diverse captions.

In the second part, we focus on how to effectively edit inputs to generate new texts for specific natural language generation tasks, e.g., text style transfer and textual adversarial example generation. For the text style transfer task, we first examine the limitations and drawbacks of current generative models for text style transfer tasks with limited data. We then develop domain adaptive text style transfer models to leverage massively available data from other domains to solve the scarce data issue in the target domain. To generate textual adversarial examples, while previous rule-based editing methods are agnostic to the input context, we propose a contextualized perturbation approach to generate fluent and grammatical adversaries with better textual similarity. We further investigate three different perturbations to construct a richer range of generation strategies, resulting in a higher attack success rate of generated adversaries.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

First and foremost, my greatest thanks go to my advisor Professor Ming-Ting Sun. I am so grateful that Professor Sun accepted me as a Ph.D. student in his group in 2016, when I was struggling in transferring my major from Physics. Over the years under his guidance and mentoring, I have learned how to think critically and independently, how to conduct research, and how to properly write academic papers. I am also very appreciative for his unfailing patience and support, giving me the freedom to explore the research direction I am interested in.

I would also like to express my great thanks to Dr. Xiaodong He, who served as my co-advisor in my early Ph.D. stage. Xiaodong directed me to the research field of natural language processing. His support is essential for my initial exploration of my Ph.D. research topics. I would also like to express my appreciation for Professor Linda Shapiro and Professor Yejin Choi for their time and effort to serve in my committee. I had served as TAs multiple times for Linda's and Yejin's courses, which are unforgettable and invaluable experience to me.

I would like to deliver my sincere gratitude to Drs. Yizhe Zhang, Zhe Gan, Yu Cheng, Lei Zhang, Chris Brockett and Bill Dolan at Microsoft Research, where I have participated in enlightening discussions and valuable collaborations during my internship. I also deeply thank Dr. Peng Wang for hosting me as a research intern at Google Research, and Drs. Li Deng, Pusheng Zhang and Yu Liu for hosting me as a research intern at Citadel.

Besides, I have been extremely lucky to be surrounded by many great friends and people, who keep supporting me in my graduate career. Among many others, this includes Hao Peng, Kevin Lin, Maolong Tang, Jun Xie, Haoming Chen, Shumo Chu, Zhitao Zhang, Yao

Lu, Tongshuang Wu, Lianhui Qin, Yuchen Jin, Tianyi Zhou, Danyang Zhuo, Shengjie Jin, Anran Wang, Ji He, Zhijie Zhou, Zhuoyu Dong, Anni Ji, Brenda Larson, Maria Vii, Kevin Smith, Josh Mutch, Charlie Fieseler, Eris Vera Machado, Joshua Sanchez from University of Washington, Liqun Chen, Guoyin Wang, Shuyang Dai from Duke University, ChengCheng Yi, Xinyu Shen, Yi Xiong, Jia Li, Yuanhui Tang, Bin Xiao, Boyi Li, Chang Li, Qing Huang, Zhouhan Lin, Yichen Gong and Yingyi Luo.

DEDICATION

to my mother, Yanqin Huang, and my father, Ke Li, for their unconditional love.

Chapter 1

INTRODUCTION

Artificial Intelligence (AI) has been influencing human lives and bringing great conveniences for human society. One of the most important goals for AI is to develop intelligent agents, which can understand human languages and help us to convey messages more efficiently. Building Generative models for natural language generation is one of the most promising approaches toward this goal. It typically follows two steps: learning and inference. The learning step endows the model to approximate the underlying distribution of the observed dataset, such as a corpus. The learned model, on the other hand, can generate novel data subjected to the observed data distribution in the inference step. Due to the discrete nature of linguistic representations, effective quantitative evaluations of generative models on the natural language tasks are non-trivial. To reflect the desired attributes of generated samples, certain quantitative metrics, such as BLEU [97], CIDEr [134], METEOR [7] or task-specific metrics [94, 120], have been proposed to fit the expectation of different language generation tasks.

Benefiting from recent advancements of deep learning techniques, deep generative models have achieved remarkable progresses in many natural language generation tasks, such as machine translation [6, 150, 133, 68], image captioning [26, 137, 4], style transfer [46, 124, 154] and textual adversarial examples [169, 138, 14, 57]. This is evidenced by impressive performances from state-of-the-art techniques on the existing metrics. Despite such successes, existing generative models still suffer various drawbacks in different natural language tasks. In this thesis, we will mainly investigate the limitations of current mainstream deep generative models on several natural language generation tasks, and discuss how to develop better generative models specifically for these tasks. We will first briefly provide the basic

backgrounds of these tasks and our contributions in the following parts.

1.0.1 Adversarial Ranking for Language Generation

This thesis consists of two parts. The first part explores ranking-based generative adversarial networks for natural language generation. We first proposed a novel ranking process in general generative adversarial networks for text generation. We then explore this ranking approach with generative adversarial networks in the image captioning task.

Generative Adversarial Networks for Text Generation. Generative adversarial networks (GANs) have drawn great attentions since Goodfellow *et al.* [35] introduced the framework for generating synthetic data that is similar to the real data. GAN consists of two neural network models, a discriminator D and a generator G, which compete against each other in a two-player minimax game. The discriminator aims to distinguish the synthetic data from the real data, while the generator is trained to confuse the discriminator by generating high-quality synthetic data that is as close to real data as possible. Formally, the GAN objective L can be written as:

$$\min_G \max_D L(G, D) = \mathop{\mathbb{E}}_{x \sim \mathcal{P}_{data}} [\log D(x)] + \mathop{\mathbb{E}}_{z \sim \mathcal{P}_z} [\log(1 - D(G(z))], \tag{1.1}$$

where $\mathbb{E}$ is the expectation operator, $\mathcal{P}_{data}$ is the real data distribution, $\mathcal{P}_z$ a latent distribution where the generator G samples synthetic data $G(z)$ by drawing z from it. Typically, $\mathcal{P}_z$ is defined by a standard normal distribution. During the adversarial learning, the generator minimizes the distance between the approximated probability distribution (generation distribution) $\mathcal{P}_G$ and the real data distribution $\mathcal{P}_{data}$, while the discriminator tries to maximize the distance. In this setup, the optimal discriminator is: $D^*(x) = \frac{\mathcal{P}_{data}}{\mathcal{P}_{data} + \mathcal{P}_G}$. The global minimum of the adversarial training is achieved if and only if $\mathcal{P}_G = \mathcal{P}_{data}$ and the optimal distance value is $-log4$ [35].

Although GANs have achieved great success in computer vision tasks such as image synthesis [19, 50, 69, 172, 9], there are only limited progress in natural language generation

because of the difficulty in handling discrete tokens. Specifically, the gradient of the training loss from the discriminator is estimated on discrete sequences and thus it is non-differentiable to the generator. To tackle such difficulty, two mainstream directions have been investigated over the past years. The first direction focuses on incorporating a continuous approximation of the discrete distribution on text, such as soft-argmax operator [67, 165], Gumbel-softmax trick [53, 89], and feature matching method [13]. The continuous approximation approaches make the model end-to-end differentiable, but suffer from the generated errors due to the discrepancy between learning and inference. While the alternative approaches [155, 12, 79, 42, 27] relieve this problem by adopting policy-gradient in reinforcement learning (RL) [146], they typically yield high-variance gradient estimates in the action space which affect their performance.

We argue that the issue in RL-based strategies is associated with a strong discriminator, which overly estimates the quality of sentences in a polarized way, resulting in high-variance gradients propagated to the generator. In our work, we propose a ranking-based generative adversarial network, named RankGAN, to overcome the high-variance gradient problem and generate high-quality language descriptions. By viewing a set of data samples collectively and evaluating their quality through relative ranking, the discriminator is able to make a better assessment of the quality of the samples, which in turn helps the generator to learn better. The proposed method is suitable for language learning in comparison to conventional GANs with a common binary discriminator.

Generative Adversarial Networks for Diverse Image Captioning. Image captioning is one of the most important applications in both computer vision and natural language processing fields, which aims to automatically generate natural descriptions for images. The task requires models to understand the content of images and then verbalize the details with natural language. The generative models G for image captioning often consist of a combination of image recognition and language generation parts. With the recent surge of deep learning techniques, the visual features of images are captured by a deep convolution

neural network (CNN), and the descriptions are generated by a deep autoregressive model implemented by a long short-term memory network (LSTM) or its variants. Commonly, the generative models optimize the parameters via maximum likelihood estimation (MLE), i.e., maximizing the conditional log-likelihood of true descriptions $(w_1, ..., w_T)$ given the corresponding image I:

$$\min_G L(G) = -\sum_{t=1}^{T} \log\ p_G(w_t|w_1, ..., w_{t-1}, I), \tag{1.2}$$

where w_t is the t^{th} token and T is the maximum length in the sequence.

Despite phenomenal research progresses in the past several years, which is evidenced by the fact that state-of-the-art methods have already surpassed human performance on certain metrics [137, 152, 4], the machine-generated captions are expressed in a very monotonic and featureless format. While such captions are normally accurate, they often lack important characteristics in human languages - distinctiveness for each image and diversity across different images. From human perspectives, as demonstrated in [54], each image possesses its own specificity, and accordingly its related captions should acquire its distinctiveness, leading to diverse captions for different images. In general, distinctive descriptions are often pursued by a human, who can easily distinguish a specific image from a group of similar images.

Following the direction of the first part of the work, we investigate how to generate captions that are not only accurate in describing an image but also diverse across different images by using a ranking-based conditional generative adversarial network. Specifically, instead of estimating the quality of a caption solely on one image, we propose a comparative adversarial learning framework that can better assess the quality of captions by comparing a set of captions within the image-caption joint space. By contrasting with human-written captions and image-mismatched captions, the caption generator effectively exploits the inherent characteristics of human languages, and generates more diverse captions.

1.0.2 Deep Generative Models for Text Editing

In the second part, we investigate deep generative models for text style transfer and textual adversarial example generation. Both tasks require modifications on parts of the input texts, resulting in attribute changes on the edited texts, while the task-specific text contents are preserved during the modifications.

Text Style Transfer. Text style transfer, which aims to edit an input sentence with the desired style while preserving style-irrelevant content, has received increasing attention in recent years. It has been applied successfully to stylized image captioning [29], personalized conversational response generation [158], formalized writing [109], offensive to non-offensive language transfer [24], and other stylized text generation tasks [1, 166].

Text style transfer has been explored as a sequence-to-sequence learning task using parallel datasets [55]. Parallel dataset denotes that each sentence expressed in one style in the dataset is annotated with a corresponding sentence written in another different style. However, parallel datasets are often not available, and hand-annotating sentences in different styles is expensive. Consequently, most previous text style transfer works consider a more realistic setting when only non-parallel stylized corpora are available. Such task is coined as unsupervised text style transfer [154]. This makes the text style transfer even more challenging because the style and content in the natural language are difficult to be disentangled without supervised signals from parallel data. The recent surge of deep generative models [64, 35] has spurred progresses in text style transfer without parallel data by learning disentanglement [46, 124, 28, 75, 102]. However, these methods typically require massive amounts of data [126], and may perform poorly in limited data scenarios.

In this task, we explore the deep generative model in text style transfer with data-scarcity issue. We show that most of previous works with the non-parallel corpora, despite of substantial progresses, yield poor performance where the generated texts tend to use the most discriminative stylized words that the target style prefers while ignoring the necessary content. To solve this issue, we examine domain adaptive methods for text style transfer

with non-parallel data to leverage massively available data from other domains. We propose simple yet effective domain adaptive text style transfer models, enabling domain-adaptive information exchange. The proposed models presumably learn from the source domain to: (*i*) distinguish stylized information and generic content information; (*ii*) maximally preserve content information; and (*iii*) adaptively transfer the styles in a domain-aware manner. We evaluate the proposed models on two style transfer tasks (sentiment and formality) over multiple target domains where only limited non-parallel data is available. Extensive experiments demonstrate the effectiveness of the proposed model compared to the baselines.

Textual Adversarial Example. A textual adversarial example modifies an input sentence and is supposed to trigger an error by the victim machine learning model. At the same time, the textual modifications to the input sentence should be minimal, such that the adversary is close to the original sentence, and the human predictions on the example remain unchanged. Besides exposing the systems' vulnerabilities and thus helping improve their robustness and security [169, 138, 14, 57], adversarial examples can also be used to interpret the models' decisions [56, 116].

In computer vision applications, minor perturbations to continuous pixels can be barely perceptible to humans, and thus one can hardly distinguish the adversarial example and its input image [38]. It is not the case for text, however, since changes to the discrete tokens are more likely to be noticed by humans. To sustain enough similarity between the adversary and its input sentence, an adversarial example generator can be built on a similarity module, e.g., synonym substitution, to control the distance with a textual similarity constraint. Nevertheless, textual similarity is not able to capture the quality of adversarial examples comprehensively. For example, randomly shuffling a few tokens in a text may sustain the semantic meaning and mislead a model, while it breaks other basic properties, fluency and grammaticality, in natural language. Therefore, besides the attacking demand and textual similarity constraint, a good textual adversarial example requires to contain reasonable fluency and grammaticality properties.

In this task, we take a step forward to build a contextualized perturbation approach. The proposed approach perturbs the input text with a masking-then-infilling procedure. By leveraging the knowledge from the pretrained masked language model, our model carries out perturbations in a context-aware manner, which prevents inappropriateness in the perturbed text. While previous token replacement methods fail to generate grammatical and fluent adversaries, our model maximally preserves the semantic meaning, fluency and grammaticality of adversarial examples. Meanwhile, our model breaks the constraint of prefixed substitution rules with diverse perturbation operations, it can search over a significantly larger space of attacking possibilities to achieve a higher attack success rate.

1.0.3 Contributions

To summarize, the contributions of this report are as follows:

- Investigating limitations of current deep generative models on several natural language tasks and provide novel and better models to overcome the limitations.

- In Chapter 2, we propose a generic and efficient ranking-based generative adversarial network on text generation task that overcomes the high-variance training problem and generates high-quality descriptions.

- In Chapter 3, we apply the ranking-based generative adversarial network on the image captioning task, generating more diverse and better captions for images by ranking captions in a collection.

- In Chapter 4, we explore a challenging domain adaptation problem for text style transfer by leveraging massively-available data from other domains to solve the data-scarcity issue. We propose two simple yet effective domain adaptation models for text style transfer.

- In Chapter 5, we investigate the task of textual adversarial attack. We propose a contextualized adversarial example generation model to generate high-quality textual adversaries in terms of attack success rate, textual similarity, fluency and grammaticality.

Chapter 6 summarizes this thesis and discusses future research works.

Part I

ADVERSARIAL RANKING FOR LANGUAGE GENERATION

Chapter 2

RANKING-BASED GENERATIVE ADVERSARIAL NETWORK

2.1 Introduction

Language generation plays an important role in natural language processing, which is essential to many applications such as machine translation [6], image captioning [26], and dialogue systems [115]. Recent studies [39, 44, 128, 150] show that the recurrent neural networks (RNNs) and the long short-term memory networks (LSTMs) can achieve impressive performances for the task of language generation. Evaluation metrics such as BLEU [97], METEOR [7], and CIDEr [134] are reported in the literature.

Generative adversarial networks (GANs) have drawn great attentions since Goodfellow *et al.* [35] introduced the framework for generating the synthetic data that is similar to the real data. The main idea behind GANs is to have two neural network models, the discriminator and the generator, competing against each other during learning. The discriminator aims to distinguish the synthetic data from the real data, while the generator is trained to confuse the discriminator by generating high quality synthetic data. During learning, the gradient of the training loss from the discriminator is used as the guidance for updating the parameters of the generator. Since then, GANs have achieved great performance in many computer vision tasks including image synthesis [19, 50, 69, 106, 117]. Their successes are mainly attributed to training the discriminator to estimate the statistical properties of the continuous real-valued data (e.g., pixel values).

The adversarial learning framework provides a possible way to synthesize language descriptions in high quality. However, GANs have limited progress with natural language processing. Primarily, the GANs have difficulties in dealing with discrete data (e.g., text

sequences [8]). In natural languages processing, the text sequences are evaluated as discrete tokens whose values are non-differentiable. Therefore, the optimization of GANs is challenging. Secondly, most of the existing GANs assume the output of the discriminator to be a binary predicate indicating whether the given sentence is written by human or machine [18, 67, 74, 153, 155]. For a large variety of natural language expressions, this binary predication is too restrictive, since the diversity and richness inside the sentences are constrained by the degenerated distribution due to the binary classification.

In this chapter, we propose a novel adversarial learning framework, RankGAN, for generating high-quality language descriptions. RankGAN learns the model from the relative ranking information between the machine-written and the human-written sentences in an adversarial framework. In the proposed RankGAN, we relax the training of the discriminator to a learning-to-rank optimization problem. Specifically, the proposed new adversarial network consists of two neural network models, a generator and a ranker. As opposed to performing a binary classification task, we propose to train the ranker to rank the machine-written sentences lower than human-written sentences with respect to a reference sentence which is human-written. Accordingly, we train the generator to synthesize sentences which confuse the ranker so that machine-written sentences are ranked higher than human-written sentences in regard to the reference. During learning, we adopt the policy gradient technique [131] to overcome the non-differentiable problem. Consequently, by viewing a set of data samples collectively and evaluating their quality through relative ranking, the discriminator is able to make better assessment of the quality of the samples, which in turn helps the generator to learn better. Our method is suitable for language learning in comparison to conventional GANs. Experimental results clearly demonstrate that our proposed method outperforms the state-of-the-art methods.

2.2 Related Work

Generative Adversarial Networks. Recently, GANs [35] have been widely explored due to its nature of unsupervised deep learning. Though GANs have achieved great successes

on computer vision applications [19, 50, 69, 106, 117], there are only limited progresses in natural language processing because the discrete sequences are not differentiable. To tackle the non-differentiable problem, SeqGAN [155] addresses this issue by the policy gradient inspired from the reinforcement learning [131]. The approach considers each word selection in the sentence as an action, and computes the reward of the sequence with the Monte Carlo (MC) search. Their method back-propagates the reward from the discriminator, and encourages the generator to create human-like language sentences. Li *et al.* [74] apply GANs with the policy gradient method to dialogue generation. They train a Seq2Seq model as the generator, and build the discriminator using a hierarchical encoder followed by a 2-way softmax function. Dai *et al.* [18] show that it is possible to enhance the diversity of the generated image captions with conditional GANs. Yang *et al.* [153] further prove that training a convolutional neural network (CNN) as a discriminator yields better performance than that of the recurrent neural network (RNN) for the task of machine translation (MT). Among the works mentioned above, SeqGAN [155] is the most relevant study to our proposed method. The major difference between SeqGAN [155] and our proposed model is that we replace the regression based discriminator with a novel ranker, and we formulate a new learning objective function in the adversarial learning framework. In this condition, the rewards for training our model are not limited to binary regression, but encoded with relative ranking information.

Learning to rank. Learning to rank plays an essential role in Information Retrieval (IR) [83]. The ranking technique has been proven effective for searching documents [48] and images [98]. Given a reference, the desired information (such as click-through logs [59]) is incorporated into the ranking function which aims to encourage the relevant documents to be returned as early as possible. While the goal of previous works is to retrieve relevant documents, our proposed model takes the ranking scores as the rewards to learn the language generator. Our proposed RankGAN is one of the first generative adversarial network which learns by relative ranking information.

2.3 Proposed Model

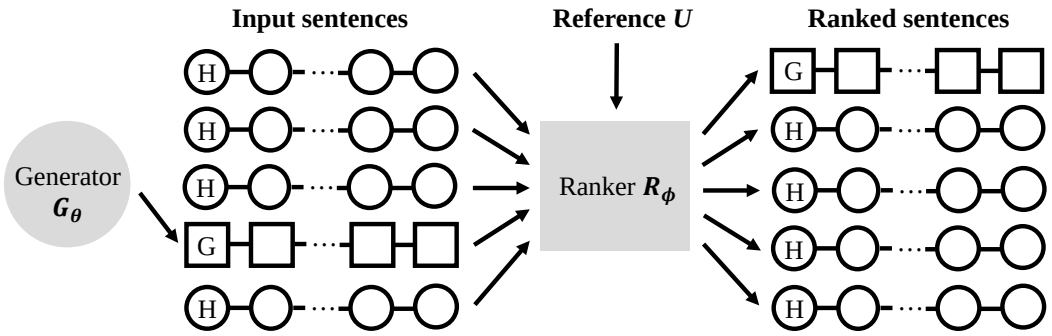


Figure 2.1: An illustration of the proposed RankGAN. **H** denotes the sentence sampled from the human-written sentences. **G** is the sentence generated by the generator G_θ. The inputs of the ranker R_ϕ consist of one synthetic sequence and multiple human-written sentences. Given the reference sentence U which is written by human, we rank the input sentences according to the relative scores. In this figure, it is illustrated that the generator tries to fool the ranker and let the synthetic sentence to be ranked at the top with respect to the reference sentence.

2.3.1 Overall architecture

In conventional GANs [35], the discriminator with multilayer perceptrons outputs a binary probability distribution to suggest whether the unknown sequences come from the real data rather than the data synthesized by a generator. In contrast to conventional GANs, RankGAN consists of a sequence generator G_θ and a ranker R_ϕ, where R_ϕ can endow a relative rank among the sequences when given a reference. As illustrated in Figure 2.1, the learning objective of G_θ is to produce a synthetic sentence **G** that receives higher ranking score than those drawn from real data **H**. However, the goal of R_ϕ is to rank the synthetic sentence **G** lower than human-written sentences **H**. Figure 2.1 illustrates that the generator

tries to fool the ranker and let the synthetic sentence $\mathbf{G}$ to be ranked at the top with respect to the reference sentences $\mathbf{U}$. Thus, this can be treated as G_θ and R_ϕ play a minimax game with the objective function $\mathfrak{L}$:

$$\min_{\theta}\max_{\phi}\mathfrak{L}(G_\theta, R_\phi) = \mathop{\mathbb{E}}_{s\sim\mathcal{P}_h}\left[\log R_\phi(s|U,\mathcal{C}^-)\right] + \mathop{\mathbb{E}}_{s\sim G_\theta}\left[\log(1-R_\phi(s|U,\mathcal{C}^+))\right] \tag{2.1}$$

where θ and ϕ are the variable parameters in G and R, respectively. $\mathbb{E}$ is the expectation operator, and $\mathcal{P}_h$ is the real data from human-written sentences. $s \sim \mathcal{P}_h$ and $s \sim G_\theta$ denote that s is from human-written sentences and synthesized sentences, respectively. U is the reference set used for estimating relative ranks, and $\mathcal{C}^+, \mathcal{C}^-$ are the comparison set with regard to different input sentences s. When the input sentence s is the real data, $\mathcal{C}^-$ contains generated data pre-sampled from G_θ; If the input sentence s is the synthetic data, the human-written data is pre-sampled and enclosed in $\mathcal{C}^+$.

The forms of G_θ and R_ϕ can be achieved in many ways. In this chapter, we design the generative model with the long short-term memory networks (LSTMs) [44]. A LSTM iteratively takes the embedded features of the current token w_t plus the information in the hidden state h_{t-1} and the cell state c_{t-1} from previous stages, and updates the current states h_t and c_t. Additionally, the subsequent word w_{t+1} is conditionally sampled subjects to the probability distribution $p(w_{t+1}|h_t)$ which is determined by the value of the current hidden state h_t. Benefiting from the capacity of LSTMs, our generative model can conserve long-term gradient information and produce more delicate word sequences $s = (w_0, w_1, w_2, ..., w_T)$, where T is the sequence length.

Recent studies show that the convolutional neural network can achieve high performance for machine translation [32, 153] and text classification [160]. The proposed ranker R, which shares the similar convolutional architecture, first maps concatenated sequence matrices into the embedded feature vectors $y_s = \mathfrak{F}(s)$ through a series of nonlinear functions $\mathfrak{F}$. Then, the ranking score will be calculated for the sequence features y_s with the reference feature y_u which is extracted by R in advance.

2.3.2 Rank score

More disparities between sentences can be observed by contrasts. Inspired by this, unlike the conventional GANs, our architecture possesses a novel comparison system that evaluates the relative ranking scores among sentences. Inspired by ranking steps commonly used in Web search [48], we formulate a relevance score of the input sequence s given a reference u by:

$$\alpha(s|u) = cosine(y_s, y_u) = \frac{y_s \cdot y_u}{\|y_s\|\|y_u\|} \tag{2.2}$$

where the y_u and y_s are the embedded feature vectors of the reference and the input sequence, respectively. $\|\cdot\|$ denotes the norm operator. Then, a softmax-like formula is used to compute the ranking score for a certain sequence s given a comparison set $\mathcal{C}$ (In Figure 2.1, s is the generated sentence $\mathbf{G}$, and $\mathcal{C}$ includes all human-written sentences $\mathbf{H}$):

$$P(s|u, \mathcal{C}) = \frac{exp(\gamma\alpha(s|u))}{\sum_{s' \in \mathcal{C}'} exp(\gamma\alpha(s'|u))} \tag{2.3}$$

The parameter γ, whose value is set empirically during experiments, shares the similar idea with the Boltzmann exploration [129] method in reinforcement learning. Lower γ results in all sentences to be nearly equiprobable, while higher γ increases the biases toward the sentence with the greater score. The set $\mathcal{C}' = \mathcal{C} \cup \{s\}$ denotes the set of input sentences to be ranked.

Since the reference sentence u is not available in most tasks, we use human-written sentences to serve the reference space. To reduce the reference variance, the collective ranking score for an input sentence is an expectation of its scores given different references sampled across the reference space. During learning, we randomly sample a set of references from human-written sentences to construct the reference set U. Meanwhile, the comparison set $\mathcal{C}$ will be constructed according to the type of the input sentence s, i.e., $\mathcal{C}$ is sampled from the human-written set and machine-generated set. With the above setting, the expected ranking

score computed for the input sentence s can be derived by:

$$R_\phi(s|U,\mathcal{C}) = \mathop{\mathbb{E}}_{u\in U}[P(s|u,\mathcal{C})] \tag{2.4}$$

Here, s is the input sentence. It is either human-written or produced by G_θ. Accordingly, u is a reference sentence sampled from set U. Given the reference set and the comparison set, we are able to compute the rank scores indicating the relative ranks for the complete sentences. The ranking scores will be used for the objective functions of generator G_θ and ranker R_ϕ.

2.3.3 Training

In conventional settings, GANs are designed for generating real-valued image data and thus the generator G_θ consists of a series of differentiable functions with continuous parameters guided by the objective function from the discriminator D_ϕ [35]. Unfortunately, the synthetic data in the text generation task is based on discrete symbols, which are hard to update through common back-propagation. To solve this issue, we adopt the Policy Gradient method [131], which has been widely used in reinforcement learning.

Suppose the vocabulary set is V, at time step t, the previous tokens generated in the sequence are $(w_0, w_1, ..., w_{t-1})$, where all tokens $w_i \in V$. When compared to the typical reinforcement learning algorithms, the existing sequence $s_{1:t-1} = (w_0, w_1, ..., w_{t-1})$ is the current state, the next token w_t selected in the next step is an action sampled from the policy $\pi_\theta(w_t|s_{1:t-1})$. Since we use G_θ to generate the next token, the policy π_θ equals to $p_G(w_t|s_{1:t-1})$, which is the conditional probability of w_t given $s_{1:t-1}$ in the generation, and θ is the parameter set in generator G. Once the generator reaches the end of one sequence (i.e., $s = s_{1:T}$), it receives a ranking reward $R(s|U,\mathcal{C})$ according to the comparison set $\mathcal{C}$ and its related reference set U.

Note that in reinforcement learning, the current reward is compromised by the rewards from intermediate states and future states. However, in text generation, the generator G_θ

obtains the reward if and only if one sequence has been completely generated, which means no intermediate reward is gained before the sequence hits the end symbol. To relieve this problem, we utilize the Monte Carlo rollouts methods [18, 155] to simulate intermediate rewards when a sequence is incomplete. Then, the expected future reward V for partial sequences can be computed by:

$$V_{\theta,\phi}(s_{1:t-1}, U) = \mathop{\mathbb{E}}_{s_r \sim G_\theta} \left[R_\phi(s_r | U, \mathcal{C}^+, s_{1:t-1}) \right] \tag{2.5}$$

Here, s_r represents the complete sentence sampled by rollout methods with the given starter sequence $s_{1:t-1}$. To be more specific, the beginning tokens $(w_0, w_1, ..., w_{t-1})$ are fixed and the rest tokens are consecutively sampled by G_θ until the last token w_T is generated. We denote this as the "path" generated by the current policy. We keep sampling n different paths with the corresponding ranking scores. Then, the average ranking score will be used to approximate the expected future reward for the current partial sequence.

With the feasible intermediate rewards, we can finalize the objective function for complete sentences. Refer to the proof in [131], the gradient of the objective function for generator G can be formulated as:

$$\nabla_\theta \mathfrak{L}_\theta(s_0) = \mathop{\mathbb{E}}_{s_{1:T} \sim G_\theta} \left[\sum_{t=1}^{T} \sum_{w_t \in V} \nabla_\theta \pi_\theta(w_t | s_{1:t-1}) V_{\theta,\phi}(s_{1:t}, U) \right] \tag{2.6}$$

where ∇_θ is the partial differential operator. The start state s_0 is the first generated token w_0. $\mathbb{E}_{s_{1:T} \sim G_\theta}$ is the mean over all sampled complete sentences based on current generator's parameter θ within one minibatch. Note that we only compute the partial derivatives for θ, as the R_ϕ is fixed during the training of generator. Importantly, different from the policy gradients methods in other works [18, 81, 155], our method replaces the simple binary outputs with a ranking system based on multiple sentences, which can better reflect the quality of the imitate sentences and facilitate effective training of the generator G.

To train the ranker's parameter set ϕ, we can fix the parameters in θ and maximize Equation (2.1). In practice, however, it has been found that the network model learns better

by minimizing $\log(R_\phi(s|U, \mathcal{C}^+))$ instead of maximizing $\log(1 - R_\phi(s|U, \mathcal{C}^+))$, where $s \sim G_\theta$. This is similar to the finding in [110]. Hence, during the training of R_ϕ, we maximize the following ranking objective function:

$$\mathfrak{L}_\phi = \mathop{\mathbb{E}}_{s\sim\mathcal{P}_h}\left[\log R_\phi(s|U, \mathcal{C}^-)\right] - \mathop{\mathbb{E}}_{s\sim G_\theta}\left[\log R_\phi(s|U, \mathcal{C}^+)\right] \tag{2.7}$$

It is worthwhile to note that when the evaluating data come from the human-written sentences, the comparison set $\mathcal{C}^-$ should consist of the generated sentences through G_θ; In contrast, if the estimating data to be ranked belongs to the synthetic sentences, $\mathcal{C}^+$ should consist of human-written sentences. We found empirically that this gives more stable training.

2.3.4 Discussion

Note that the proposed RankGAN has a Nash Equilibrium when the generator G_θ simulates the human-written sentences distribution $\mathcal{P}_h$, and the ranker R_ϕ cannot correctly estimate the rank between the synthetic sentences and the human-written sentences. However, as also discussed in the literature [35, 36], it is still an open problem how a non-Bernoulli GAN converges to such an equilibrium. In a sense, replacing the absolute binary predicates with the ranking scores based on multiple sentences can relieve the gradient vanishing problem and benefit the training process. In the following experiment section, we observe that the training converges on four different datasets, and leads to a better performance compared to previous state-of-the-arts.

2.4 Experiment

Following the evaluation protocol in [155], we first carry out experiments on the data and the simulator proposed in [155]. Then, we compare the performance of RankGAN with other state-of-the-art methods on multiple public language datasets including Chinese poems [162], COCO captions [80], and Shakespear's plays [123].

2.4.1 Simulation on synthetic data

We first conduct the test on the dataset proposed in [155]. The synthetic data[1] is a set of sequential tokens which can be seen as the simulated data comparing to the real-word language data. We conduct this simulation to validate that the proposed method is able to capture the dependency of the sequential tokens. In the simulation, we firstly collect $10,000$ sequential data generated by the oracle model (or true model) as the training set. Note that the oracle model we used is a random initialized LSTM which is publicly available[1]. During learning, we randomly select one training sentence and one generated sentence from RankGAN to form the input set $\mathcal{C}'$. Then, given a reference sample which is also randomly selected from the training set, we compute the ranking score and optimize the proposed objective function. Note that the sentence length of the training data is fixed to 20 for simplicity.

Following the evaluation protocol in [155], we evaluate the machine-written sentences by stimulating the Turing test. In the synthetic data experiment, the oracle model, which plays the role as the human, generates the *"human-written"* sentences following its intrinsic data distribution $\mathcal{P}_o$. We use these sentences as the ground truth sentences used for training, thus each model should learn and imitate the sentences from $\mathcal{P}_o$. At the test stage, obviously, the generated sentences from each model will be evaluated by the original oracle model. Following this, we take the sentences generated by RankGAN as the input of the oracle model, and estimate the average negative log-likelihood (NLL) [49]. The lower the NLL score is, the higher probability the generated sentence will pass the Turing test.

We compare our approach with the state-of-the-art methods including maximum likelihood estimation (MLE), policy gradient with BLEU (PG-BLEU), and SeqGAN [155]. The PG-BLEU computes the BLEU score to measure the similarity between the generated sentence and the human-written sentences, then takes the BLEU score as the reward to update

[1]The synthetic data and the oracle model (LSTM model) are publicly available at https://github.com/LantaoYu/SeqGAN

Method	MLE	PG-BLEU	SeqGAN	RankGAN
NLL	9.038	8.946	8.736	**8.247**

Table 2.1: Performance comparison of different methods on the synthetic data in terms of the negative log-likelihood (NLL) scores.

the generator with policy gradient. Because PG-BLEU also learns the similarity information during training, it can be seen as a baseline comparing to our approach. It's noteworthy that while the PG-BLEU grasps the similarities depend on the n-grams matching at the token-level among sentences, RankGAN explores the ranking connections inside the embedded features of sentences. These two methods are fundamentally different. Table 2.1 shows the performance comparison of RankGAN and the other methods. It can be seen that the proposed RankGAN performs more favourably against the compared methods. Figure 2.2 shows the learning curves of different approaches with respect to different training epochs. The vertical dashed line indicates the end of the pre-training of PG-BLEU, SeqGAN and RankGAN. While MLE, PG-BLEU and SeqGAN tend to converge after 200 training epochs, the proposed RankGAN consistently improves the language generator and achieves relatively lower NLL score. The results suggest that the proposed ranking objective, which relaxes the binary restriction of the discriminator, is able to learn effective language generator. It is worth noting that the proposed RankGAN achieves better performance than that of PG-BLEU. This indicates employing the ranking information as the reward is more informative than making use of the BLEU score that stands on token-level similarities. In our experiments, we noticed that the results are not sensitive to the size of comparison set and reference set. The learning curves converge to similar results with different reference sizes and comparison sizes. However, learning with the large reference size and comparison set could potentially increase the computational cost.

Conventional GANs employ a binary classifier to distinguish the human-written and

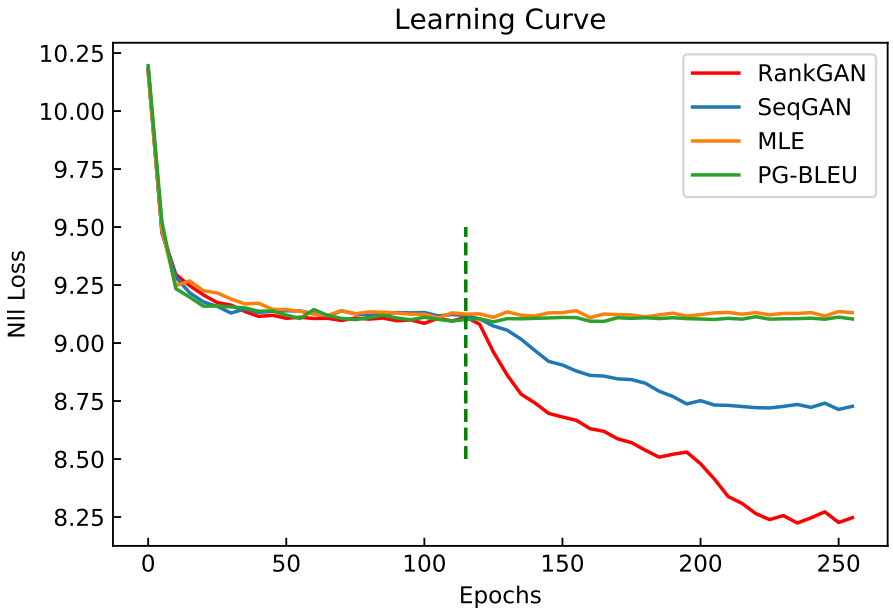


Figure 2.2: Learning curves of different methods on the simulation of synthetic data with respect to different training epochs. Note that the vertical dashed line indicates the end of the pre-training of PG-BLEU, SeqGAN and RankGAN.

the machine-created sentences. Though effective, it is also very restrictive for tasks like natural language generation, where rich structures and various language expressions need to be considered. For these tasks, usually a relative quality assessment is more suitable. The proposed RankGAN is able to perform quality assessment in a relative space, and therefore, rather than training the discriminator to assign the absolute 0 or 1 binary predicate to the synthesized or real data sample, we expect the discriminator to rank the synthetic data compared to the real data in the relative assessment space where better quality judgments of different data samples can be obtained. Given the rewards with the relative ranking information, the proposed RankGAN is possible to learn a better language generator than the compared state-of-the-art methods.

Method	BLEU-2	Method	Human score
MLE	0.667	SeqGAN	3.58
SeqGAN	0.738	RankGAN	4.52
RankGAN	**0.812**	Human-written	**6.69**

Table 2.2: The performance comparison of different methods on the Chinese poem generation in terms of the BLEU scores and human evaluation scores.

2.4.2 Results on Chinese poems composition

To evaluate the performance of our language generator, we compare our method with other approaches including MLE and SeqGAN [155] on the real-word language data. We conduct experiments on the Chinese poem dataset [162], which contains $13,123$ five-word quatrain poems. Each poem has 4 sentences, and each sentence contains 5 words resulting in a total of 20 words. After the standard pre-processing which replaces the non-frequently used words (appeared less than 5 times) with the special character *UNK*, we train our model on the dataset and generate the poem. To keep the proposed method general, our model does not take advantage of any prior knowledge such as phonology during learning.

Following the evaluation protocol in [155, 162], we compute the BLEU-2 score and estimate the similarity between the human-written poem and the machine-created one. Table 2.2 summarizes the BLEU-2 score of different methods. It can be seen that the proposed RankGAN performs more favourably compared to the state-of-the-art methods in terms of BLEU-2 score. This indicates that the proposed objective is able to learn effective language generator with real-world data.

We further conduct human study to evaluate the quality of the generated poem in human perspective. Specifically, we invite 57 participants who are native mandarin Chinese speakers to score the poems. During the evaluation, we randomly sample and show 15 poems written by different methods, including RankGAN, SeqGAN, and written by human. Then, we ask

Method	BLEU-2	BLEU-3	BLEU-4
MLE	0.781	0.624	0.589
SeqGAN	0.815	0.636	0.587
RankGAN	**0.845**	**0.668**	**0.614**

Method	Human score
SeqGAN	3.44
RankGAN	4.61
Human-written	**6.42**

Table 2.3: Performance comparison of different methods on the COCO captions in terms of the BLEU scores and human evaluation scores.

the subjects to evaluate the quality of the poem by grading the poem from 1 to 10 points. It can be seen in Table 2.2, human-written poems receive the highest score comparing to the machine-written one. RankGAN outperforms the compared method in terms of the human evaluation score. The results suggest that the ranking score is informative for the generator to create human-like sentences.

2.4.3 Results on COCO image captions

We further evaluate our method on the large-scale dataset for the purpose of testing the stability of our model. We test our method on the image captions provided by the COCO dataset [80]. The captions are the narrative sentences written by human, and each sentence is at least 8 words and at most 20 words. We randomly select $80,000$ captions as the training set, and select $5,000$ captions to form the validation set. We replace the words appeared less than 5 times with *UNK* character. Since the proposed RankGAN focuses on unconditional GANs that do not consider any prior knowledge as input, we train our model on the captions of the training set without conditioning on specific images.

In the experiment, we evaluate the performance of the language generator by averaging BLEU scores to measure the similarity between the generated sentences and the human-written sentences in the validation set. Table 2.3 shows the performance comparison of different methods. RankGAN achieves better performance than the other methods in terms

Human-written
Two men happily working on a plastic computer.
The toilet in the bathroom is filled with a bunch of ice.
A bottle of wine near stacks of dishes and food.
A large airplane is taking off from a runway.
Little girl wearing blue clothing carrying purple bag sitting outside cafe.
SeqGAN (Baseline)
A baked mother cake sits on a street with a rear of it.
A tennis player who is in the ocean.
A highly many fried scissors sits next to the older.
A person that is sitting next to a desk.
Child jumped next to each other.
RankGAN (Ours)
Three people standing in front of some kind of boats.
A bedroom has silver photograph desk.
The bears standing in front of a palm state park.
This bathroom has brown bench.
Three bus in a road in front of a ramp.

Table 2.4: Example of the generated descriptions with different methods. Note that the language models are trained on COCO caption dataset without the images.

Method	**BLEU-2**	**BLEU-3**	**BLEU-4**
MLE	0.796	0.695	0.635
SeqGAN	0.887	0.842	0.815
RankGAN	**0.914**	**0.878**	**0.856**

Table 2.5: Performance comparison of different methods on Shakespeare's play - *Romeo and Juliet* in terms of the BLEU scores.

of different BLEU scores. Some of the samples written by humans, and synthesized by the SeqGAN and the proposed model RankGAN are shown in Table 2.4. These examples show that our model is able to generate fluent, novel sentences that are not existing in the training set, and RankGAN is able to learn effective language generator in a large corpus.

We also conduct human study to evaluate the quality of the generated sentences. We invite 28 participants who are native or proficient English speakers to grade the sentences. Similar to the setting in previous section, we randomly sample and show 15 sentences written by different methods, and ask the subjects to grade from 1 to 10 points. Table 2.3 shows the human evaluation scores. As can be seen, the human-written sentences get the highest score comparing to the language models. Among the GANs approaches, RankGAN receives better score than SeqGAN, which is consistent to the finding in the Chinese poem composition. The results demonstrate that the proposed learning objective is capable to increase the diversity of the wording making it realistic toward human-like language description.

2.4.4 Results on Shakespeare's plays

Finally, we investigate the possibility of learning Shakespeare's lexical dependency, and make use of the rare phrases. In this experiment, we train our model on the Romeo and Juliet play [123] to further validate the proposed method. The script is splited into $2,500$ training sentences and 565 test sentences. To learn the rare words in the script, we adjust the

threshold of *UNK* from 5 to 2. Table 2.5 shows the performance comparison of the proposed RankGAN and the other methods including MLE and SeqGAN. As can be seen, the proposed method achieves significantly higher BLEU score than the other methods in terms of different n-grams criteria. The results indicate the proposed RankGAN is able to capture the transition pattern among the words, even if the training sentences are novel, delicate and complicated.

2.5 Conclusion

We presented a new generative adversarial network, RankGAN, for generating high-quality natural language descriptions. Instead of training the discriminator to assign absolute binary predicate to real or synthesized data samples, we propose using a ranker to rank the human-written sentences higher than the machine-written sentences relatively. We then train the generator to synthesize natural language sentences that can be ranked higher than the human-written one. By relaxing the binary-classification restriction and conceiving a relative space with rich information for the discriminator in the adversarial learning framework, the proposed learning objective is favorable for synthesizing natural language sentences in high quality. Experimental results on multiple public datasets demonstrate that our method achieves significantly better performance than previous state-of-the-art language generators.

Chapter 3

COMPARATIVE ADVERSARIAL LEARNING FOR DIVERSE IMAGE CAPTIONING

3.1 Introduction

Image caption generation has attracted great attentions due to its wide applications in many fields, such as semantic image search, image commenting in social chat bot, and assistance to visually impaired people. Benefiting from recent advancements of deep learning, most existing works employ convolutional neural networks (CNNs) and deep recurrent language models, and have achieved great performance improvement on automatic evaluation metrics, such as BLEU [97], CIDEr [134], etc.

Despite such successes, machine-generated captions are often in a generic format and can be easily differentiated from human-written captions, which tend to be more descriptive and diverse. As most state-of-the-art image caption algorithms are learning-based, to best match with the ground truth captions, such algorithms often produce high-frequency n-gram patterns or common expressions. As a result, the generated image captions receive high scores on automatic evaluation metrics, yet lack a significant characteristic in human language - diversity across different images. From human perspectives, as demonstrated in [54], each image possesses its own specificity, and accordingly its related captions should acquire its distinctiveness, leading to diverse captions for different images. In general, distinctive descriptions are often pursued by human, who can easily distinguish a specific image among a group of similar images. In this chapter, our goal is to generate diverse and accurate captions which are similar to human-written descriptions.

Recent success of Generative Adversarial Networks (GANs) [93] provides a possible way to generate diverse captions [18, 125]. In this setting, a caption generator and a discriminator

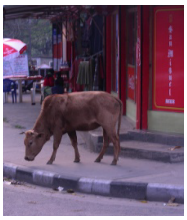
Images	Generated Captions	Binary Score	Comparative Score
	**MLE:** a man wearing a suit and tie	0.15	0.45
	G-GAN: a young man in business gear poses for the camera	0.18	0.59
	CAL (ours): a man with glasses wearing a striped tie and black suit	0.19	0.72
GT: a man with glasses and his eyes closed dressed in a black shirt and a necktie			
	**MLE:** a cow standing on the side of a street	0.50	0.73
	G-GAN: a brown cow standing in a city	0.49	0.68
	CAL (ours): a large cow walking on a side street in front of a door	0.52	0.82
GT: there is a cow on the sidewalk standing in front of a door			

Figure 3.1: Captions generated by MLE, conditional GANs (G-GAN) with binomial scores and our comparative adversarial learning network (CAL) with comparative scores. The shown scores are evaluated by the discriminators in G-GAN and CAL, respectively. The proposed adversarial framework estimates comparative scores by comparing a collection of captions, leading to more accurate and discriminative rewards for caption generator.

are jointly trained by a binomial distribution, which estimates the relevance and quality of the captions to the image. However, due to the large variability of natural language, a binary predictor is usually incapable of representing the richness and diversity of captions. To ensure semantic relevance, a regularization term for distinguishing mismatched captions must be included during training.

In contrast to assigning an absolute score to a caption for one image, we noticed that it is relatively easier to distinguish the qualities of two captions by comparison. Motivated by this, we propose a comparative adversarial learning (CAL) network to learn human-like captions. Specifically, contrary to an absolute binary score for one caption, the quality of the caption is assessed relatively by comparing it with other captions in the image-caption space.

In adversarial learning, the proposed discriminator ranks the human references, which are more specific and distinctive, higher than generic captions that have high-frequency n-gram patterns or common expressions. Consequently, with the guides from the discriminator, the generator effectively learns to generate more specific and distinctive captions, hence increases the diversity across the corpus. In summary, our main contributions lie in three aspects:

- We propose a novel comparative adversarial learning network, which is capable of generating more diverse and better captions across images by comparing different captions.

- By suppressing the scores of image-mismatched captions, especially for those from similar images, the proposed comparative learning framework can inherently ensure semantic relevance without involving an regularization term for mismatched captions.

- To effectively measure the caption diversity across images, we propose a new metric based on the semantic variance from caption embedding features. Additionally, experimental results clearly demonstrate the effectiveness of the proposed framework in terms of diversity and quality.

3.2 Related Work

Diverse Image Captioning. Most image captioning systems use an encoder-decoder framework which shares a similar idea as sequence learning [128, 32, 133, 101, 100]. Typically, the networks are trained by maximum likelihood estimation (MLE) [137, 62, 152, 30] or reinforcement learning [114, 113, 82, 4, 87, 84]. Although such methods achieve outstanding performances on conventional evaluation metrics, such as BLEU, CIDEr, etc., the generated captions usually consist of high-frequency n-gram patterns but lose the diversity across images and thus are unnatural to human. To remedy this weakness, diverse beam search and ensemble methods [135, 143] have been proposed. [141, 11] work on diverse image captioning by using variational auto-encoders. To achieve better caption diversity, [18, 125] incorporate generative adversarial networks (GANs) into image captioning systems, with a binary-based

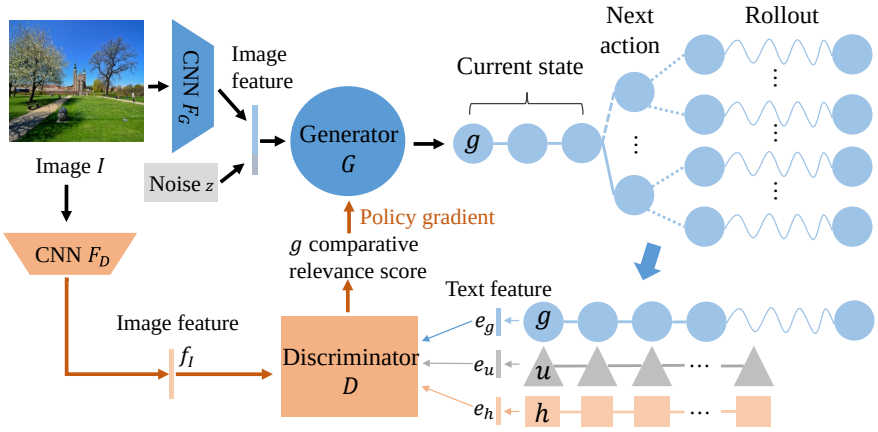


Figure 3.2: Comparative Adversarial Learning Network. The discriminator D is trained over comparative relevance scores for each image by comparing a generated caption g, a human-written caption h, and unrelated captions u. The generator G is optimized by policy gradient where the reward is estimated by the expectation of the comparative relevance score g over K rollout simulation captions.

discriminator. However, in sequence adversarial training, a binary-based discriminator is easily trained much stronger than the generator [12, 42, 79], resulting in less distinguishable rewards or gradient vanishing problems for the generator (Figure 3.1). Our proposed adversarial framework estimates comparative scores by comparing a collection of captions, leading to more accurate and discriminative rewards for caption generator. To generate captions with correct semantic relevance, [18, 125] must train the binary discriminator under an additional regularization.

In this chapter, we propose a comparative adversarial learning framework that explicitly estimates the quality of captions in a more discriminative way, which in turn helps the generator to produce more diverse captions while maintaining the caption correctness without the additional discriminator regularization.

Diversity Metrics. Automatic evaluation metrics such as BLEU, CIDEr-D, etc., have been widely applied for evaluating the quality of generated captions. Nonetheless, the evaluation of diversity across captions is still an open problem. Human language, inherited immense complexity and sophisticated interpretation, poses a thorny problem for developing standard criterion. [73, 135, 52] measure the degree of diversity by analyzing distinct n-grams or word usages for generated sentences with respect to ground truths. This reflects an inventiveness for generated sentences, but not a diversity aspect among all the generated sentences. To estimate the caption diversity at the token level, [125, 141, 20] inspect n-gram usage statistics and the size of vocabulary in all generate captions. However, the diversity of sentences is not only represented by various word or phrase usages, but also variant long-term sentence patterns and even implications of sentences. A simple n-gram statistics is unable to assess the diversity at the sentence level. In this chapter, we propose a novel diversity metric based on semantic sentence features which compensate the defects of previous methods.

3.3 Proposed Model

As shown in Figure 3.2, the proposed Comparative Adversarial Learning (**CAL**) Network consists of a caption generator G and a comparative relevance discriminator (**cr-discriminator**) D. The two subnetworks play a min-max game as follow:

$$\min_{\theta} \max_{\phi} \mathcal{L}(G_\theta, D_\phi), \tag{3.1}$$

in which $\mathcal{L}$ is an overall loss function, while θ and ϕ are trainable parameters in G and D, respectively. Given a reference image I, the generator G_θ outputs a sentence g as the corresponding caption. D_ϕ aims at correctly estimating the comparative relevance score (**cr-score**) of g with respect to human-written caption h within the image-caption joint space. G_θ is trained to maximize the cr-score of g and generate human-like descriptions trying to confuse D_ϕ. We will elaborate each subnetwork in the following sections.

3.3.1 Caption Generator

Our caption generator G_θ is based on the standard encoder-decoder architecture [137]. The captioning image encoder model F_G first extracts a fixed dimensional feature from image I using a CNN. Then a text decoder, implemented by a long short-term memory (LSTM) network, interprets the encoded feature $F_G(I)$ into a word sequence $g_{0:T} = (g_0, ..., g_T)$ to describe image I, where g_t is a token in time step t and T is the maximum time step. To produce captions with more variations, the input feature $F_G(I)$ can be varied by concatenating with a random vector z. The notation of z will be ignored in the rest parts for simplicity. In time step t generation, the next token g_t can be sampled by:

$$g_t \sim \pi_\theta(g_t|I, g_{0:t-1}), \quad t \in (1, T). \tag{3.2}$$

π_θ is a word distribution, determined by inputs and θ, over all the words in vocabulary V. By sequentially sampling or greedy decoding words according to π_θ, a complete caption $g_{1:T}$ can be generated by captioner G_θ. In comparative adversarial training, G_θ expects to produce better captions with higher cr-scores. However, unlike cross-entropy loss in the MLE method, the cr-score of $g_{1:T}$ estimated by discriminator D_ϕ is based on discrete tokens, whose gradients cannot be directly employed for G_θ through back-propagation. Therefore, we adopt a common technique - Policy Gradient method [130] to solve this gradient issue. The details will be discussed in Section 3.3.3.

3.3.2 Comparative Relevance Discriminator

Dai *et al.*[18] propose to estimate the semantic relevance, naturalness, and quality of a generated caption by a logistic function over the similarity between the caption and the given image. However, an absolute binary value is very restrictive to evaluate them all, especially the quality of a caption. To evaluate a discriminative score, it is more justifiable to compare a generated caption with other captions, primarily with human-written caption h. Therefore, we formulate a comparative relevance score (**cr-score**) to measure an overall

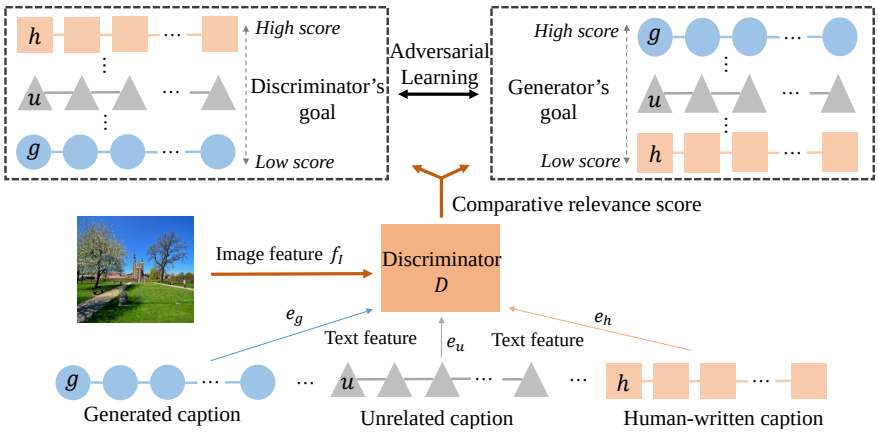


Figure 3.3: Training objectives in our adversarial learning. While the discriminator desires to judge human-written captions correctly with higher cr-scores, the generator aims to produce captions with higher cr-scores and thus confusing the discriminator.

image-text quality of caption c by comparing a set of captions $\mathcal{C}^c$ given image I:

$$D_\phi(c|I,\mathcal{C}^c) = \frac{exp(\gamma S(e_c, f_I)}{\sum_{c' \in \mathcal{C}^c} exp(\gamma S(e_{c'}, f_I))}, \tag{3.3}$$

where $\mathcal{C}^c$ denotes a set of captions including c, and the cr-score of c is what we care about here. e_c and f_I are the text feature and image feature extracted by the text encoder and CNN image encoder F_{D_ϕ} in discriminator D_ϕ, respectively. The cosine similarity between e_c and f_I is defined as $S(e_c, f_I) = (e_c^T f_I)/(\|e_c\|\|f_I\|)$. $\|\|$ is the $L2$ Euclidean norm. γ is an empirical parameter defined by validation experiment. A higher γ leads $D_\phi(c|I,\mathcal{C}^c)$ towards the caption that better matches with image I.

$D_\phi(c|I,\mathcal{C}^c))$ estimates the cr-score of caption c by comparing with other captions in the image-caption joint space - a higher score represents caption c is superior in $\mathcal{C}^c$. To obtain more accurate cr-score for c, it is favorable to include human-written caption h for image I in $\mathcal{C}^c$. In this case, the cr-score of c contains a discrepancy information between caption c and human-written caption h. The discriminator is designed to differentiate generated captions from human-written captions for image I. Specifically, from the discriminator's perspective,

a human-written caption desires to receive a higher cr-score, whereas a generated caption should receive a lower cr-score (Figure 3.3). Hence, the objective function to be maximized for discriminator can be defined as:

$$\mathop{\mathbb{E}}_{h\sim\mathcal{P}_h}\left[\log D_\phi(h|I,\mathcal{C}^h)\right]+\mathop{\mathbb{E}}_{g\sim G_\theta}\left[\log(1-D_\phi(g|I,\mathcal{C}^g)\right], \tag{3.4}$$

where $\mathcal{P}_h(I)$ represents human-written caption distribution given image I. Set $\mathcal{C}^h$ and $\mathcal{C}^g$ encloses a human-written caption h, a machine-generated caption g, and other unrelated captions u. In experiments, u can be directly obtained from image-mismatched captions in one mini-batch.

3.3.3 Policy Gradient Optimization for G_θ

In contrast to D_ϕ, the caption generator G_θ attempts to maximize the cr-scores of machine-generated captions and thus fool the discriminator (Figure 3.3). However, the cr-scores of a generated caption g are assessed by D_ϕ based on a series of sequential discrete samples, which are non-differentiable during training. We address this problem by a classic policy gradient method [130]. Considering in each time step t, the generation of each word g_t is an action of an "agent" G_θ from policy π_θ according to the current state $(I, g_{0:t-1})$. An intermediate reward r for this action is approximated as the expected future reward:

$$Q_\theta(g_t|I,g_{0:t-1})=\mathop{\mathbb{E}}_{g_{t+1:T}}[r(g_{0:t-1},g_t,g_{t+1:T}|I)]. \tag{3.5}$$

The action reward r can be any metric, including the cr-score from D_ϕ. Unfortunately, the discriminator cannot provide a score unless a complete sentence is generated. The lack of intermediate rewards will result in a gradient vanishing problem. To imitate an accurate intermediate reward, following [155], we deploy a K-times Monte Carlo rollout process conditioned on the current caption generator G_θ to explore the rest unknown words $g_{t+1:T}$. Then the intermediate reward for action g_t can be approximated by the expected cr-score over K

rollout simulation captions:

$$Q_{\theta,\phi}(g_t|I, g_{0:t-1}) \simeq \frac{1}{K}\sum_{k=1}^{K} D_\phi(g_{k,t}|I, C^{g_{k,t}}), \tag{3.6}$$

where $g_{k,t} = g_{0:t-1}, g_t \oplus \bar{g}_{t+1,T}$, and $\bar{g}_{t+1,T}$ is sampled from G_θ by the rollout method. $\oplus$ is the token concatenation operator. Besides, $t \in (1, T-1)$, as g_0 is a start token and G_θ receives an accurate reward once generating a full sequence. $C^{g_{k,t}}$ contains a simulated caption $g_{k,t}$, a human-written caption h and other unmatched descriptions u, corresponding to the image I. To train the generator, the objective is to optimize the policy and adjust the generator to receive a maximum long-term reward - higher cr-scores for generated captions in each time step (Figure 3.3). In the end, the gradient for updating generator G_θ can be finalized by:

$$\mathop{\mathbb{E}}_{g\sim G_\theta} \sum_{t=1}^{T} \nabla_\theta \pi_\theta(g_t|I, g_{0:t-1}) \cdot Q_{\theta,\phi}(g_t|I, g_{0:t-1}), \tag{3.7}$$

where g_t is an intermediate token belonging to g at time step t. The goal of the generator is to maximize the expected cr-scores of generated captions.

3.3.4 Comparisons with Previous Models

During discriminator training, [18, 125] introduce a regularization term to learn image-caption matching by minimizing binary scores of mismatched captions u (last term in the below equation):

$$\mathop{\mathbb{E}}_{h,g,u} \log D_b(h|I) + \log(1 - D_b(g|I)) + \log(1 - D_b(u|I)), \tag{3.8}$$

where D_b is a binary discriminator. However, the cr-discriminator D_ϕ can naturally learn such image-caption matching by placing mismatched captions in the comparison set C^h with true captions h and generated captions g. Specifically, by enlarging the cr-score of the matched image-caption pair (h, I) in set C^h, D_ϕ can consistently distinguish its corresponding caption from others, and suppress the scores for mismatched descriptions u (Equation (3.3)). D_ϕ can in turn help the caption generator G_θ produce diverse captions for corresponding

images, ensuring semantic relevances of generated captions. Meanwhile, the binary discriminator D_b separates the decisions on g and h. The proposed network simply combines the two separate decisions into a single ranking process. The cr-score of the generated captions are estimated by contrasting human-written sentences subject to image I. This can assist the cr-score to comprise more informative guidances, including both naturalness and quality from ground truths, benefiting the training of the caption generator G_θ.

3.4 Experiment

Models. To test the effectiveness of the proposed Comparative Adversarial Learning (CAL) network, we compare two baseline models:

- **MLE**: We use LSTM-R [30] based on the mainstream CNN-LSTM architecture as our MLE baseline model. The training follows the standard MLE method.
- **Adversarial models**: We use *G-GAN* [18] as the baseline model for diverse image captioning (**G** represents the generator). The corresponding discriminator D_b outputs a binary score in $[0, 1]$ through a logistic function over the dot product between image and text features (Equation (3.8)).

To make a fair comparison, all image features for generators and discriminators are extracted by *ResNet-152* [43] (we reimplement *G-GAN* by using *ResNet-152* network as image encoders). Following [62], we convert all the captions to lowercases and remove its non-alphabet characters. We also discard the tokens with frequency less than 5 in the training dataset, resulting in a vocabulary size of 8,791. Both image encoders F_{G_θ} and F_{D_ϕ} in the generator and discriminator are implemented using *ResNet* [43] with 152 layers, separately. The image activations in the *pool5* layer are extracted, yielding 2048-dimensional image features. Noise vector z with 100-dimensions is sampled from a uniform distribution. All the image features are projected to 512 dimensions by fully connected layers. The text-decoder in the generator and the text-encoder in the discriminator are all implemented using LSTMs with 512 hidden nodes. We use the last hidden activations from the text-encoder as text feature, which shares the same dimension with the projected image feature.

Training. Before adversarial training, the caption generator G_θ in both adversarial models is pretrained by the standard MLE method [137] [62] for 20 epochs, and the cr-discriminator is pretrained according to Equation (3.4) for 10 epochs. During the experiment, we found the generator pretraining is necessary, otherwise it will encounter mode collapse problem and generate nonsense captions. On the other hand, pretraining discriminator helps more stable training later. In the adversarial learning stage, two sub-networks are trained jointly, in which every one generator iteration is followed by 5 discriminator iterations. We set the learning rate to 0.0005 and the batch size to 64. In each mini-batch with 64 image-caption pairs, all other 63 captions that are not corresponding to the correct image are used as the unrelated captions during training. The rollout number K is empirically set to 16, and γ is set to 10. During testing, the generated captions are sampled based on policy and the one with the best cr-score is chosen for evaluation.

Dataset. We conduct all experiments on the MSCOCO dataset [80]. MSCOCO contains 123,287 images, each being annotated with at least 5 human-written captions. All our experiments are based on the public split method from [62]: 5000 images for both validation and testing, and the rest for training.

Evaluation. We evaluate the generated captions based on both the correctness and diversity metrics, which guarantee the generation quality in each aspect. While the correctness of the generated captions is measured by common captioning metrics (e.g., BLEU [97], CIDEr [134], etc), the diversity across various images is evaluated by the proposed metric based on caption embedding features.

Consider each image is annotated by one caption, whose embedding feature is extracted by a same text encoder. Ideally, all embedding features are identical and the feature variance is zero if all the images have same captions. Conversely, a large variance would present if all the captions were distinct. Thus, the variance across embedding features reflects the diversity of captions on a semantic-level. To measure the variance, all the text embedding

Model	BLEU-4	METEOR	ROUGE	CIDEr	SPICE
Human	0.190	0.240	0.465	0.861	**0.208**
MLE	**0.297**	**0.252**	**0.519**	**0.921**	0.175
G-GAN	0.208	0.224	0.467	0.705	0.156
CAL (ours)	0.213	0.225	0.472	0.721	0.161

Table 3.1: Performance comparisons on MSCOCO test set. In human result, a sentence randomly sampled from ground-truth annotations is evaluated by the rest annotations for each image.

features can be concatenated into a feature matrix $A \in \mathbb{R}^{m \times n}$, where m is the number of captions and n is the dimensions of the embedding feature. To estimate the correlation σ_i in each dimension, the covariance matrix $M \in \mathbb{R}^{n \times n}$ of A can be computed. Then, σ_i can be obtained by singular value decomposition (SVD): $M = U\Sigma V^T$, where $\Sigma = diag(\sigma_0, ..., \sigma_{n-1})$; U and V^T are $m \times m$ and $n \times n$ unitary matrix.

Finally, we use l_1-norm $\hat{\sigma} = \sum_{i=0}^{n-1} |\sigma_i|$ to evaluate an overall variance in all dimensions among caption embedding features. A large variance $\hat{\sigma}$ suggests the embedding features of captions are less akin or correlated, representing more distinctive expressions and larger diversity among image captions.

3.4.1 Accuracy

We first evaluate the generated captions from different models on five automatic metrics: BLEU4 [97], METEOR [7], ROUGE_L [78], CIDEr-D [134] and SPICE [3]. As can be seen in Table 3.1, although our method CAL slightly outperforms the baseline G-GAN, the standard MLE model yields remarkably better results, even outperforms human. However,

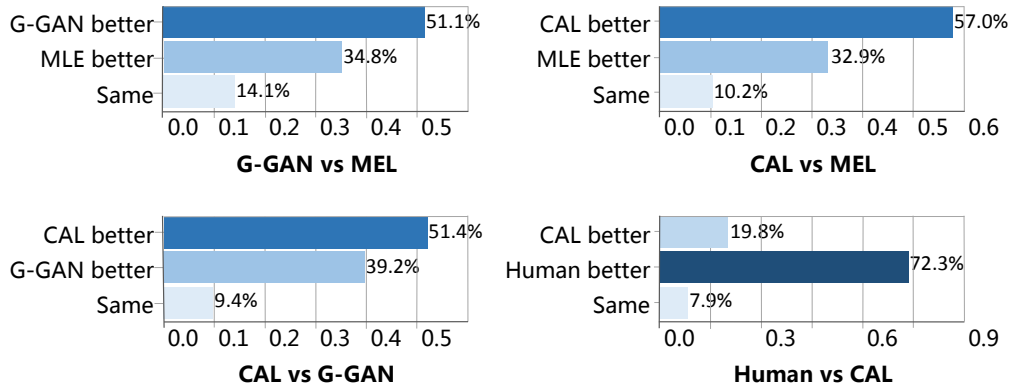


Figure 3.4: Human evaluation results by comparing model pairs. The majority of respondents agree that our proposed CAL generates better captions than the two baselines. The numbers in the figure represent the ratio of total survey cases.

as discussed by [18, 125], these evaluation metrics overly focus on n-grams matching with ground truth captions and ignore other important human language factors such as diversity. The captions, written with variant expressions, have fewer n-grams matched with ground truths. As a result, captions with novel expressions from the human and adversarial models receive lower scores on these metrics. These metrics particularly represent the quality of pattern matching, instead of the overall quality from human perspective.

3.4.2 Human Evaluation

To correlate with human judgments on the correctness of captions, we conducted human evaluation experiments on Amazon Mechanical Turk. Specifically, we randomly sampled 300 images from test set. Then, given an image with two generated captions from different models, subjects are asked to choose one caption that best describes the image. We received more than 9000 responses in total and the results are summarized in Figure 3.4. The numbers in the figure represent the ratio of total survey cases. It can be seen that the majority of people consider the captions from G-GAN and especially our CAL better than those from the standard MLE method. This illustrates that despite both adversarial models perform

Category	**MLE**	**G-GAN**	**CAL(ours)**	**Human**
Bathroom	2.733	6.145	**6.501**	9.066
Computer	3.710	6.012	**7.228**	8.943
Pizza	3.837	5.779	**6.805**	9.117
Building	4.019	5.940	**6.088**	9.344
Cat	4.196	5.225	**6.473**	9.155
Car	4.968	5.910	**6.661**	8.741
Daily supply	5.056	6.204	**7.330**	9.075
All Categories	6.947	7.759	**8.812**	9.465

Table 3.2: Diversity evaluations across various image categories.

poorly on automatic metrics, the generated captions are of higher quality in terms of human views. In the comparison between CAL and G-GAN, our model can generates more human-like captions that receive more acknowledgements. This demonstrates that, by exploiting more comparative relevance information against ground truth and other captions instead of solely on image, the proposed CAL effectively improves the caption generator and achieves better captions.

3.4.3 Diversity

To compare the capabilities of generating diverse expressions, we measure the variances of captions from different models across images. All the embedding features are extracted using the same text-encoder in our framework. Besides estimating the variance across all the images in the test set, we also inspect the variances inside different image categories. Particularly, We use the K-means method to cluster the input image features, and select six clusters with high-frequency topics. All the results are summarized in Table 3.2. It can be seen that despite the MLE method performs well on automatic metrics, the variance of

captions is relatively lower across different images. As shown in Figure 3.5, the MLE model often generates similar expressions and meanings within one category, even if the images are distinct.

In contrast to the MLE model, both adversarial models, especially our proposed CAL, can generate more diverse captions with respects to distinct images. G-GAN uses a binary discriminator to separates the decisions on machine-generated and human-written captions. Compared to G-GAN, our network trained by comparative learning binds the information of human-written captions which possess highest diversity characteristics as indicated in Table 3.2. The comparative learning also encourages the distinctiveness of the generated captions by suppressing the cr-scores of mismatched captions, especially for those from akin images. These allow our caption generator to produce more descriptive captions for different images. As expected, the variance of captions from our model is larger than that from G-GAN across all the images. Similar trends can be observed inside different categories. This suggests that our proposed CAL has better generative capability than the baseline G-GAN and helps bridge the gap between machine-generated and human-written captions. Figure 3.6 shows that the CAL model is able to generate diverse captions for each image.

3.4.4 Ablation Study

We study the diversity effect of each component in our network on MSCOCO val set. The results are summarized in Table 3.3, where we can see that the sampling decoding and noise vectors bring a certain amount of diversity. The proposed comparative relevance discriminator compares different captions and maximizes the scores of the generated captions among a set of references, resulting in an even larger diversity gain.

3.4.5 Network Effectiveness

We further investigate the effectiveness of adversarial models by a caption-image matching experiment [18, 17]. Specifically, if all the generated captions have enough diversity and the adversarial discriminator is good enough to distinguish related and unrelated image-caption

Bathroom				
MLE	a bathroom with a toilet and a sink	a bathroom with a toilet and a sink	a bathroom with a sink and a mirror	a bathroom with a sink and a toilet
G-GAN	a bathroom with a white toilet and tiled walls	a restroom with a toilet sink and shower	a bathroom with a white bathtub and two sinks and a mirror	a pink restroom with a toilet inside of it
CAL	a toilet sits inside of a bathroom next to a wall	a narrow bathroom with a toilet sink and a shower with dirty walls	a clean bathroom with a large sink bathtub and a mirror	a pink bathroom with a sink toilet and mirror
Pizza				
MLE	a pizza sitting on top of a white plate	a pizza sitting on top of a white plate	a close up of a pizza on a table	a pizza sitting on top of a pan
G-GAN	a pizza on a plate on a wooden table	a pizza sitting on a plate next to a glass of wine	the pizza is covered with cheese and tomatoes	a close up of a sliced pizza on a plate
CAL	a cheese pizza on a plate sits on a table	a plate of pizza and a glass of beer on the table	a pizza topped with lots of toppings is ready to be cut	a partially eaten pizza is being cooked on a pan
Car				
MLE	a green truck parked in a parking lot	a black truck is parked in a parking lot	a group of buses driving down a street	a city bus stopped at a bus stop
G-GAN	a green garbage truck in a business district	an antique black car sitting in a parking lot	a city street filled with taxis and buses	people are waiting in line as the bus travel down the road
CAL	a large green truck driving past a tall building	an old style truck parked in a parking space near a building	the city buses are driving through the traffic	people gather to a street where a bus get ready to board

Figure 3.5: Qualitative results illustrate that adversarial models, especially our proposed CAL, can generate more diverse descriptions.

				
z_1	a man holding a rope while skateboarding a small wave	a baseball game is in progress with the large crowd watching	twin skiers stand on skis on a snowy hillside	a crowd of people are having a meal and drinking beer at a table
z_2	a person is trying to flip while skiing water	a person in jersey holding a baseball bat in swing position	two people in ski gear standing on a snowy slope	a group of people are eating outside at a restaurant
z_3	there is a male surfer that is riding a wave	a batter catcher and umpire during a baseball game	two skiers on a thick snow covered mountain	many people are sitting on tables at a restaurant

Figure 3.6: Qualitative results of images captions generated by comparative adversarial learning network with different random vector z.

Adversarial Model				
Beam search	**Sampling**	**Noise**	**Comparative learning**	**Diversity**
✓				7.078
	✓			7.331
	✓	✓		7.784
	✓	✓	✓	**8.845**

Table 3.3: Ablation study of caption diversity of our adversarial model. **Beam search** and **Sampling** indicate beam search and sampling decoding respectively. **Noise** denotes adding noise vectors in decoding. **Comparative learning** represents our discriminator with comparative learning.

pairs, the corresponding image could be easily retrieved by the discriminator when given its own caption. For each adversarial model, we can use its generated caption as a query to rank

Model	**R@1**	**R@3**	**R@5**	**R@10**
MLE	3.03	8.67	12.75	20.54
G-GAN	16.07	33.66	43.80	59.74
G-GAN *w/o reg.*	14.24	30.28	40.11	56.13
CAL (ours)	**18.81**	**36.56**	**46.84**	**62.57**

Table 3.4: Caption-image retrieval comparison evaluated on MSCOCO test set. Captions are all self-generated by each model. *G-GAN w/o reg.* denotes the G-GAN model without the regularization term in the discriminator. The recall ratio is calculated by ranking discriminator's scores based on caption-image pairs.

all images, based on the similarity scores from corresponding discriminators. Then a recall ratio can be calculated by inspecting the top-k resulting images in the ranked list. Since the MLE model is not an adversarial model, we use the discriminator from G-GAN to retrieve the generated captions, providing a baseline for comparison.

We summarize the performance comparison in Table 3.4. Captions are all self-generated by each model. Although captions from MLE commonly describe images well, they are less diverse for different images, resulting in a poor retrieval performance. Meanwhile, our proposed CAL outperforms all the other models, including the adversarial model G-GAN. This further demonstrates that CAL can produce more diverse captions for all images. In Table 3.4, it is noteworthy that the G-GAN model needs an regularization term to sustain better semantic relevance of captions. Without such regularization, our CAL model still improves discernibility on caption-image pairs. It proves that the cr-discriminator in our proposed network can provide more accurate rewards during adversarial training, leading to a better caption generator.

3.5 Conclusion

We presented a comparative adversarial learning network for generating diverse captions across images. A novel comparative learning schema is proposed for the discriminator, which better assesses the quality of captions by comparing with other captions. Thus more caption properties including correctness, naturalness, and diversity can be taken into consideration. This in turn benefits the caption generator to effectively exploit inherent characteristics inside human languages and generate more diverse captions. We also proposed a new caption diversity metric in the semantic level across images. Experimental results clearly demonstrate that our proposed method generates better captions in terms of both accuracy and diversity across images.

Part II

TEXT EDITING FOR LANGUAGE GENERATION

Chapter 4

DOMAIN ADAPTIVE TEXT STYLE TRANSFER

4.1 Introduction

Text style transfer, which aims to edit an input sentence with the desired style while preserving style-irrelevant content, has received increasing attention in recent years. It has been applied successfully to stylized image captioning [29], personalized conversational response generation [158], formalized writing [109], offensive to non-offensive language transfer [24], and other stylized text generation tasks [1, 166].

Text style transfer has been explored as a sequence-to-sequence learning task using parallel datasets [55]. Parallel dataset denotes that each sentence expressed in one style in the dataset is annotated with a corresponding sentence written in another different style. However, parallel datasets are often not available, and hand-annotating sentences in different styles is expensive. The recent surge of deep generative models [64, 35] has spurred progress in text style transfer without parallel data by learning disentanglement [46, 124, 28, 75, 102]. These methods typically require massive amounts of data [126], and may perform poorly in limited data scenarios.

A natural solution to the data-scarcity issue is to resort to massive data from other domains. However, directly leveraging abundant data from other domains is problematic due to the discrepancies in data distribution on different domains. Different domains generally manifest themselves in domain-specific lexica. For example, sentiment adjectives such as "*delicious*", "*tasty*", and "*disgusting*" in restaurant reviews might be out of place in movie reviews, where the sentiment words such as "*imaginative*", "*hilarious*", and "*dramatic*" are more typical. Domain shift [40] is thus apt to result in feature misalignment.

In this work, we take up the problem of domain adaptation in scenarios where the target

domain data is scarce and misaligned with the distribution in the source domain. Our goal is to achieve successful style transfer into the target domain, with the help of the source domain, while the transferred sentences carry relevant characteristics in the target domain.

We present two first-of-their-kind domain adaptive text style transfer models that facilitate domain-adaptive information exchange between the source and target domains. These models effectively learn generic content information and distinguish domain-specific information. Generic content information, primarily captured by modeling a large corpus from the source domain, facilitates better content preservation on the target domain. Meanwhile, domain-specific information, implicitly imposed by domain vectors and domain-specific style classifiers, underpins the transferred sentences by generating target-specific lexical terms.

Our contributions in this paper are threefold: (*i*) We explore a challenging domain adaptation problem for text style transfer by leveraging massively-available data from other domains. (*ii*) We introduce simple text style transfer models that preserve content and meanwhile translate text adaptively into target-domain-specific terms. (*iii*) We demonstrate through extensive experiments the robustness of these methods for style transfer tasks (sentiment and formality) on multiple target domains where only limited non-parallel data is available.

4.2 Related Work

Text Style Transfer. Text style transfer using neural networks has been widely studied in the past few years. A common paradigm is to first disentangle latent space as content and style features, and then generate stylistic sentences by tweaking the style-relevant features and passing through a decoder. [46, 28, 124, 154, 34, 79] explored this direction by assuming the disentanglement can be achieved in an auto-encoding procedure with a suitable style regularization, implemented by either adversarial discriminators or style classifiers. [75, 151, 164] achieved disentanglement by filtering the stylistic words of input sentences. Recently, [102] has proposed to use back-translation for text style transfer with a de-noising auto-encoding objective [86, 126]. Our work differs in that we leverage domain adaptation to deal

with limited target domain data, whereas previous methods require massive target domain style-labelled samples.

Domain Adaptation. Domain adaptation has been studied in various natural language processing tasks, such as sentiment classification [105], dialogue system [144], abstractive summarization [47, 163], machine translation [65, 5, 122, 90], etc. However, little or no work explores domain adaptation on text style transfer. To the best of our knowledge, we are the first to explore the adaptation of text style transfer models to a new domain with limited non-parallel data available. The task requires both style transfer and domain-specific generation on the target domain. To differentiate different domains, [121, 16] appended domain tokens to the input sentences. Our model uses learnable domain vectors combining domain-specific style classifiers, which force the model to learn distinct stylized information in each domain.

4.3 Preliminary

We first describe a standard text style transfer approach, which only considers data in the target domain. We limit our discussion to the scenario where only non-parallel data is available, since large amounts of parallel data is typically not feasible.

Given a set of style-labelled sentences $\mathcal{T} = \{(x_i, l_i)\}_{i=1}^{N}$ in the target domain, the goal is to transfer sentence x_i with style l_i to a sentence $\widetilde{x}_i$ with another style $\widetilde{l}_i$, where $\widetilde{l}_i \neq l_i$. $l_i, \widetilde{l}_i$ belong to a set of style labels $l^{\mathcal{T}}$ in the target domain: $l_i, \widetilde{l}_i \in l^{\mathcal{T}}$. Typically, an encoder encodes the input x_i to a semantic representation c_i, while a decoder controls or modifies the stylistic property and decodes the sentence $\widetilde{x}_i$ based on c_i and the pre-specific style $\widetilde{l}_i$.

Specifically, we denote an encoder-decoder model as (E, D). The semantic representation c_i of sentence x_i is extracted by the encoder E, i.e., $c_i = E(x_i)$. The decoder D aims to learn a conditional distribution of $\widetilde{x}_i$ given the semantic representation c_i and style $\widetilde{l}_i$:

$$p_D(\widetilde{x}_i|c_i, \widetilde{l}_i) = \prod_{t=1}^{T} p_D(\widetilde{x}_i^t|\widetilde{x}_i^{<t}, c_i, \widetilde{l}_i), \tag{4.1}$$

where $\widetilde{x}_i^t$ is the t^{th} token of $\widetilde{x}_i$, and $\widetilde{x}_i^{<t}$ is the prefix of $\widetilde{x}_i$ up to the t^{th} token.

Directly estimating Eqn. (4.1) is impractical during training due to a lack of parallel data $(x_i, \widetilde{x}_i)$. Alternatively, the original sentence x_i should have high probability under the conditional distribution $p_D(x_i|c_i, l_i)$. Thus, an auto-encoding reconstruction loss could be formulated as:

$$L_{ae}^{\mathcal{T}} = - \mathop{\mathbb{E}}_{x_i \sim \mathcal{T}} \log p_D(x_i|c_i, l_i)\,. \tag{4.2}$$

Note that we assume that the decoder D recovers x_i's original stylistic property as accurate as possible when given the style label l_i. To achieve text style transfer, the decoder manipulates the style of generated sentences by replacing l_i with a desired style $\widetilde{l}_i$. Specifically, the generated sentence $\widetilde{x}_i$ is sampled from $\widetilde{x}_i \sim p_D(\widetilde{x}_i|c_i, \widetilde{l}_i)$. However, by directly optimizing Eqn. (4.2), the encoder-decoder model tends to ignoring the style labels and collapses to a reconstruction model, which might simply copy the input sentence, hence fails to transfer the style. To force the model to learn meaningful style properties, [46, 45] apply a style classifier for the style regularization. The style classifier ensures the encoder-decoder model to transfer $\widetilde{x}_i$ with its correct style label $\widetilde{l}_i$:

$$L_{style}^{\mathcal{T}} = - \mathop{\mathbb{E}}_{\widetilde{x}_i \sim p_D(\widetilde{x}_i|c_i, \widetilde{l}_i)} \log P_{C^{\mathcal{T}}}(\widetilde{l}_i|\widetilde{x}_i)\,, \tag{4.3}$$

where $C^{\mathcal{T}}$ is the style classifier pretrained on the target domain. The overall training objective for text style transfer within the target domain $\mathcal{T}$ is written as:

$$L^{\mathcal{T}} = L_{ae}^{\mathcal{T}} + L_{style}^{\mathcal{T}}\,. \tag{4.4}$$

4.4 Proposed Model

In this section, we present Domain Adaptive Style Transfer (DAST) models to perform style transfer on a target domain by borrowing the strength from a source domain, while maintaining the transfer to be domain-specific.

4.4.1 Problem Definition

Suppose we have two sets of style-labelled sentences $\mathcal{S} = \{(x'_i, l'_i)\}_{i=1}^{N'}$, $\mathcal{T} = \{(x_i, l_i)\}_{i=1}^{N}$ in the source domain $\mathcal{S}$ and the target domain $\mathcal{T}$, respectively. x'_i denotes the i^{th} source sentence.

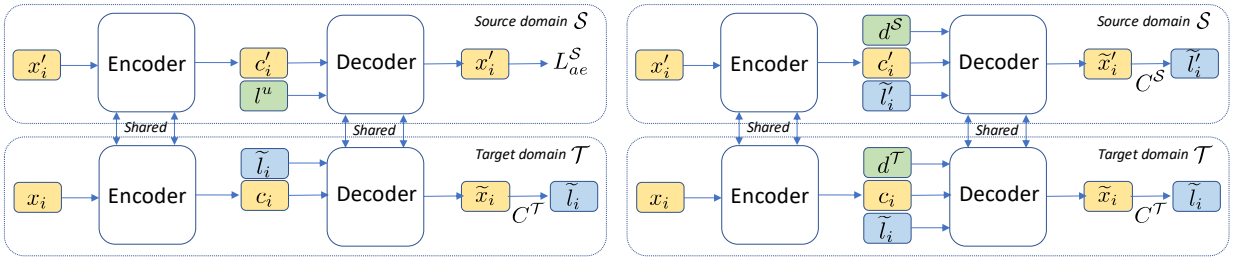


Figure 4.1: Illustration of the proposed **DAST-C** (left) and **DAST** (right) model. DAST-C learns the generic content information through $L_{ae}^{\mathcal{S}}$ on massive source domain data with unknown style l^u. For DAST, $d^{\mathcal{T}}, d^{\mathcal{S}}$ and $C^{\mathcal{T}}, C^{\mathcal{S}}$ denote domain vectors and domain-specific style classifiers, respectively. Better looked in color.

l_i' denotes the corresponding style label, which belongs to a source style label set: $l_i' \in l^{\mathcal{S}}$ (e.g., positive/negative). l_i' can be available or unknown. Likewise, pair (x_i, l_i) represents the sentence and style label in the target domain, where $l_i \in l^{\mathcal{T}}$.

We consider domain adaptation in two settings: (i) the source style $l^{\mathcal{S}}$ is unknown, e.g., we may have a large corpus, such as Yahoo! Answers, but the underlying style for each sample is not available; (ii) the source styles are available, and are the same as the target styles, i.e., $l^{\mathcal{T}} = l^{\mathcal{S}}$, e.g., both IMDB movie reviews and Yelp restaurant reviews have the same style classes (negative and positive sentiments).

In both scenarios, we assume that the target domain $\mathcal{T}$ only has limited non-parallel data. With the help of source domain data $\mathcal{S}$, the goal is to transfer (x_i, l_i) to $(\widetilde{x}_i, \widetilde{l}_i)$ in the target domain. The transferred sentence $\widetilde{x}_i$ should simultaneously hold: (i) the main content with x_i, (ii) a different style $\widetilde{l}_i$ from l_i, and (iii) domain-specific characteristics of the target data distribution $\mathcal{T}$.

4.4.2 DAST with unknown-stylized source data

In this section, we investigate the case that the source style $l^{\mathcal{S}}$ is unknown. We first examine a drawback of limited target data to motivate our method. With limited target data, Eqn. (4.4)

may yield an undesirable transferred text, where the generated text tends to use the most discriminative words that the target style prefers while ignoring the content. This is because the classifier $C^{\mathcal{T}}$ typically requires less data to train than a sequence autoencoder (E, D). The classifier objective $L^{\mathcal{T}}_{style}$ thus dominates Eqn. (4.4), rendering the generator to bias the sentences with most representative stylized (e.g., positive or negative) words rather than preserving the contents (see Table 4.4 for examples).

We consider alleviating this issue by leveraging massive source domain data to enhance the content-preserving ability, though the underlying styles in the source domain are unknown. By jointly training an auto-encoder on both the source and target domain data, the learned generic content information enables the model to yield better content preservation on the target domain.

To utilize the source data, we consider that $l^{\mathcal{S}}$ only contains a special unknown-style label l^u, separated from the target style $l^{\mathcal{T}}$. We assume the semantic representation of the source data c'_i is encoded by the encoder, i.e., $c'_i = E(x'_i)$. The decoder takes c'_i with style l^u to generate the sentences on the source domain. The auto-encoding reconstruction objective of the source domain is:

$$L^{\mathcal{S}}_{ae} = - \mathop{\mathbb{E}}_{x'_i \sim \mathcal{S}} \log p_D(x'_i | c'_i, l^u), \tag{4.5}$$

where the encoder-decoder model (E, D) is shared in both domains. Therefore, the corresponding objective can be written as:

$$L_{\text{DAST-C}} = L^{\mathcal{T}}_{ae} + L^{\mathcal{T}}_{style} + L^{\mathcal{S}}_{ae}\,. \tag{4.6}$$

This can be perceived as combining the source domain data with the target domain data to train a better encoder-decoder framework, while target-specific style information on the target domain is learned through $L^{\mathcal{T}}_{style}$.

Note that $L^{\mathcal{T}}_{ae}$ and $L^{\mathcal{S}}_{ae}$ are conditional on domain-specific styles labels: $l^{\mathcal{T}}$ and l^u, which implicitly encourages the model to learn domain-specific features. The decoder could thus generate target sentences adaptively with $l^{\mathcal{T}}$, while achieving favorable content preservation with the generic content information modeled by $L^{\mathcal{S}}_{ae}$. We refer this model, which is illustrated

in Figure 4.1(left), as *Domain Adaptive Style Transfer with generic Content preservation (DAST-C).*

4.4.3 DAST with stylized source data

We further explore the scenario where $l^{\mathcal{S}} = l^{\mathcal{T}}$. In this case, besides the generic content information, there is much style information from the source domain that could be leveraged, e.g., generic stylized expressions like "*fantastic*" and "*terrible*" for sentiment transfer can be applied to both restaurant and movie reviews. We thus consider to borrow the full strength of the source data, by sharing learned knowledge on both the generic content and style information.

A straightforward way to achieve this is to train Eqn. (4.4) on both domains. However, simply mixing the two domains together will lead to undesirable style transfers, where the transfer is not domain-specific. For example, when adapting the IMDB movie reviews to the Yelp restaurant reviews, directly sharing the style transfer model without specifying the domain will inevitably result in generations like *"The pizza is dramatic!"*.

To alleviate this problem, we introduce additional domain vectors, encouraging the model to perform style transfer in a domain-aware manner. The proposed DAST model is illustrated in Figure 4.1(right). Consider two domain vectors: $d^{\mathcal{S}}$ for the source domain and $d^{\mathcal{T}}$ for the target domain, respectively. We rewrite the auto-encoding loss as:

$$L_{ae}^{\mathcal{S},\mathcal{T}} = - \mathop{\mathbb{E}}_{x'_i \sim \mathcal{S}} \log p_D(x'_i | c'_i, d^{\mathcal{S}}, l'_i) - \mathop{\mathbb{E}}_{x_i \sim \mathcal{T}} \log p_D(x_i | c_i, d^{\mathcal{T}}, l_i)\,, \tag{4.7}$$

where the encoder-decoder model (E, D) is shared across domains. The domain vectors, $d^{\mathcal{S}}$, $d^{\mathcal{T}}$, learned from the model, implicitly guide the decoder to generate sentences with domain-specific characteristics. Note that l_i and l'_i are shared, i.e., $l^{\mathcal{T}} = l^{\mathcal{S}}$. This enables the model to learn generic style information from both domains. On the other hand, explicitly learning precise stylized information within each domain is crucial to generate domain-specific styles. Thus, two domain-specific style classifiers ensure the model to learn the corresponding styles

by conditioning on $(d^{\mathcal{S}}, \widetilde{l}'_i)$ in the source domain or $(d^{\mathcal{T}}, \widetilde{l}_i)$ in the target domain:

$$L^{\mathcal{S},\mathcal{T}}_{style} = - \mathop{\mathbb{E}}_{\widetilde{x}'_i \sim p_D(\widetilde{x}'_i|c'_i, d^{\mathcal{S}}, \widetilde{l}'_i)} \log P_{C^{\mathcal{S}}}(\widetilde{l}'_i|\widetilde{x}'_i) - \mathop{\mathbb{E}}_{\widetilde{x}_i \sim p_D(\widetilde{x}_i|c_i, d^{\mathcal{T}}, \widetilde{l}_i)} \log P_{C^{\mathcal{T}}}(\widetilde{l}_i|\widetilde{x}_i)\,, \tag{4.8}$$

where $\widetilde{x}'_i, \widetilde{x}_i$ are the transferred sentences with pre-specific styles $\widetilde{l}'_i, \widetilde{l}_i$ in the source and target domains, respectively. The domain-specific style classifiers, $C^{\mathcal{T}}$ and $C^{\mathcal{S}}$, are trained separately on each domain. The signals from classifiers encourage the model to learn domain-specific styles combining with the domain vectors and style labels. The overall training objective of the proposed DAST model is:

$$L_{\text{DAST}} = L^{\mathcal{S},\mathcal{T}}_{ae} + L^{\mathcal{S},\mathcal{T}}_{style}\,. \tag{4.9}$$

The domain-specific style classifiers enforce the model to learn domain-specific style information conditioning on $(d^{\mathcal{S}}, \widetilde{l}'_i)$ or $(d^{\mathcal{T}}, \widetilde{l}_i)$, which in turn controls the model to generate sentences with domain-specific words. The model can thus distinguish domain-specific features, and adaptively transfer the styles in a domain-aware manner.

4.5 Experiments

We evaluate our proposed models on two tasks: sentiment transfer (positive-to-negative and negative-to-positive), and formality transfer (informal-to-formal). In both tasks, we make comparisons with previous approaches over multiple target domains. All experiments are conducted on one Nvidia GTX 1080Ti GPU.

4.5.1 Dataset

A statistics for the source and target corpora used in the experiments is summarized in Table 4.1.

Sentiment Transfer. For the source domain, we use IMDB movie review corpus [22] by following the filtering and preprocessing pipelines from [124]. This results in $344k$ training samples with sentiment labels. For the target domain, both the Yelp restaurant review

Sentiment Transfer					
Source	**Train**	**Target**	**Train**	**Dev**	**Test**
IMDB	344K	Yelp	444K	4K	1K
		Amazon	554K	2K	1K
		Yahoo	4K	2K	1K
Formality Transfer					
Source	**Train**	**Target**	**Train**	**Dev**	**Test**
GYAFC	103K	Enron	6K	0.5K	0.5K

Table 4.1: Statistics of source and target datasets.

dataset and the Amazon product review dataset are from [75]. For the test sets, we evaluate our methods by using $1k$ human-transferred sentences, annotated by [75], on both Yelp and Amazon datasets. In addition to the two standard sentiment datasets, we manually collected a Yahoo sentimental question dataset - $7k$ question samples with sentiments from Yahoo! Answers dataset [161]. We split the $7k$ sentimental questions into $4k/2k/1k$ for train/dev/test sets, respectively. Note that the Yahoo sentiment dataset only consists of questions, which have different domain characteristics with the IMDB dataset. In all the sentiment experiments, we consider both transfer directions (positive-to-negative and negative-to-positive).

Formality Transfer. We use Grammarly's Yahoo Answers Formality Corpus (GYAFC) [109] as the source dataset. The publicly released version of GYAFC only covers two topics (*Entertainment & Music* and *Family & Relationships*), where each topic contains $50k$ paired informal and formal sentences written by humans. For the target domain, we use Enron email conversation dataset[1], which covers several different fields like *Business, Politics, Daily Life,*

[1] `https://www.cs.cmu.edu/~./enron/`

Style Classifier		Domain Classifier	
Dataset	**Accuracy**	**Dataset**	**Accuracy**
Yelp	97.6%	IMDB & Yelp	94.8%
Amazon	81.0%	IMDB & Amazon	97.1%
Yahoo	99.4%	IMDB & Yahoo	86.9%
ENRON	87.0%	GYAFC & ENRON	89.7%

Table 4.2: Test accuracy of evaluation classifiers.

etc. We manually labeled $7k$ *non-parallel* sentences written in either the formal or informal style. We split the Enron dataset into $6k, 500, 500$ samples for training, validation and testing, respectively. Both the validation and test set consist of mere informal sentences, where the corresponding formal references are annotated by us from a crowd-sourcing platform for evaluation. We only assess the informal-to-formal transfer direction in the formality transfer experiment.

4.5.2 Evaluation

Automatic Metrics. We evaluate the effectiveness of our DAST models based on three automatic metrics:

- **Content Preservation**: We assess the content preservation according to n-gram statistics, by measuring the BLEU scores [97] between generated sentences and human references on the target domain, refered as *human* BLEU (hBLEU). When no human reference is available (e.g., Yahoo), we compute the BLEU scores with respect to the input sentences.
- **Style Control**: We generate samples from the model and measure the style accuracy with a style classifier that is pre-trained on the target domain. We refer the style accuracy as S-acc.

	Yelp(100% data)				Amazon(100% data)			
Model	**D-acc**	**S-acc**	***h*BLEU**	**G-score**	**D-acc**	**S-acc**	***h*BLEU**	**G-score**
CrossAlign [124]	-	85.0	3.7	8.3	-	23.0	34.1	18.0
Delete&Retrieve [75]	-	90.6	14.8	17.9	-	50.9	30.3	25.7
CycleRL [151]	-	88.7	12.3	16.4	-	68.7	14.2	15.5
SMAE [164]	-	85.1	12.1	15.5	-	71.1	12.9	14.9
ControlGen [45]		91.5	25.5	27.4	-	79.0	31.1	30.5
Finetune	**96.1**	91.3	25.6	27.8	**97.4**	79.2	34.1	34.3
DAST-C (ours)	93.8	91.7	25.7	27.5	96.7	81.9	35.7	35.0
DAST (ours)	95.8	**92.3**	**26.3**	**28.9**	96.9	**83.0**	**35.9**	**35.1**
	Yelp(1% data)				**Amazon(1% data)**			
Model	**D-acc**	**S-acc**	***h*BLEU**	**G-score**	**D-acc**	**S-acc**	***h*BLEU**	**G-score**
CrossAlign [124]	-	76.3	4.8	8.5	-	83.2	2.0	5.9
Delete&Retrieve [75]	-	82.1	4.1	7.6	-	63.0	6.9	9.3
CycleRL [151]	-	86.6	1.4	5.2	-	79.5	0.7	3.8
SMAE [164]	-	96.0	1.2	4.8	-	87.2	0.4	3.2
ControlGen [45]	-	**98.5**	3.7	8.6	-	83.2	1.9	5.8
Finetune	**98.1**	96.7	13.9	18.5	**96.0**	**89.2**	11.3	14.4
DAST-C (ours)	96.9	90.3	17.8	19.3	94.8	78.2	20.1	21.6
DAST (ours)	97.0	92.6	**20.1**	**23.1**	94.6	82.7	**21.0**	**23.1**

Table 4.3: Automatic evaluation results on Yelp and Amazon test sets. D-acc and S-acc denote domain accuracy and style accuracy, respectively. G-score is the geometric mean of S-acc and *h*BLEU.

- **Domain Control**: To validate whether the generated sentences hold the characteristics of the target domain, we adopt a pre-trained domain classifier to measure the percentage of generated sentences that belong to the target domain. We refer the domain accuracy as D-acc.

All the pre-trained classifiers are implemented by TextCNN [63, 167]. After training, all classifiers are used for evaluation only. The test accuracy and domain accuracy of all these classifiers used for evaluation are reported in Table 4.2. Following [151], we also evaluate all methods using a single unified metric called G-score, which calculates the geometric mean of style accuracy and hBLEU.

Human Evaluation. To assess the quality of transferred sentences, we conduct human evaluations based on the facets of *content preservation*, *style control* and *fluency*, following [92]. We also asked each worker to provide a judgment of **the overall quality** in terms of three aspects as a whole. Previous works [126, 34] ask workers to evaluate the quality via a numerical score, however, we found that this empirically leads to high-variance results. Instead, we pair transferred sentences from two different models, and ask workers to choose the sentence they prefer *when compared to the input* on each evaluation aspect. We provide a "No Preference" option to choose when the workers think the qualities of the two sentences are indistinguishable.

For each human evaluation on Yelp sentiment transfer and Enron formality transfer tasks, we randomly sampled 100 sentences from the corresponding test set and collected three responses for each pair on every evaluation aspect, yielding 2700 responses in total. Each pair of system outputs was randomly presented to 7 crowd-sourced judges, who indicated their preference for style control, content preservation and fluency. To minimize the impact of spamming, we employed the top-ranked 30% of U.S. workers provided by the crowd-sourcing service. In order to make the task less abstract, following [92], we asked the judges to evaluate the content preservation quality independently of style information. Detailed task descriptions and examples were also provided to guide the judges. Inter-rater agreement, as

	Yelp Sentiment Transfer (positive-to-negative)
Input	the service was great , food delicious , and the value impeccable .
ControlGen	the service was *horrible* , **service** , the service and very frustrated .
Finetune	the service was *poor* , food **...** , and the **experience were** .
DAST-C	the service was *horrible* , food *horrible* , and the **slow** sparse .
DAST	the service was *horrible* , food *bland* , and the value *lousy* .
	Yelp Sentiment Transfer (negative-to-positive)
Input	and the pizza was cold , greasy , and generally quite awful .
ControlGen	and the **food** was *delicious*, delicious , and freaking tasty , delicious .
Finetune	and the pizza was professional , friendly , and always have *great* .
DAST-C	and the pizza was *fresh*, greasy , and generally *quite cool* .
DAST	and the pizza was *tasty* , *juicy* , and definitely *quite amazing* .
Human	the pizza was warm , not greasy , and generally tasted great .

Table 4.4: Transferred sentences on Yelp (1%) dataset, where *italic red* denotes successful style transfers, **bold blue** denotes content losses, and sans serif yellow denotes grammar errors. Better looked in color.

measured by agreement with the most common judgment was 75.9%.

4.5.3 Experimental Setup

The encoder E and the decoder D are implemented by one-layer GRU [15] with hidden dimensions 500 and 700, respectively. The domain-vector dimension is set to 50. The style labels are represented by learnable vectors with 150 dimensions. The decoder is initialized by a concatenation of representations of content, style, and domain vectors. If domain vectors are not used, the dimension of style labels is set to 200; accordingly, the initialization of the decoder is a concatenation of content and style representations. TextCNN [63] is employed

for the domain-specific style classifiers pre-trained on corresponding domains. After pre-training, the parameters of the classifiers are fixed. We use the hard-sampling trick [86] to back-propagate the loss through discrete tokens from the classifier to the encoder-decoder model. During training, we assign each mini-batch the same amount of source and target data to balance the training.

We make an extensive comparison with five state-of-the-art text style transfer models: CrossAlign [124], Delete&Retrieve [75], CycleRL [151], SMAE [164] and ControlGen [45]. We also experiment a simple and effective domain adaptation baseline - Finetune, which is trained with Eqn. (4.4) on the source domain and then fine-tuned on the target domain.

4.5.4 Results

Model Comparisons. To evaluate the effectiveness of leveraging massive data from other domains, we compare our proposed DAST models with previously proposed models trained on the target domain (Table 4.3). We observe that by leveraging massive data from the IMDB dataset, our models achieve better performance against all baselines on the sentiment transfer tasks in both the Yelp and Amazon domains.

Notably, when the target domain has limited data (1%), all baselines trained only on the target domain fail completely on content preservation. Finetune preserves better content but experiences the catastrophic forgetting problem [37] to the source domain information. As a result, the overall style transfer performance is still nonoptimal. By contrast, with the help of the source domain, DAST obtains considerable content preservation performance improvement when compared with other baselines. Our model also attains favorable performance in terms of style transferring accuracy (S-acc), resulting in a good overall G-score. In general, we observe that DAST-C is able to better preserve content information, while DAST further improves both content preservation and style control abilities. Additionally, both DAST-C and DAST can adapt to the target domain, as evidenced by the high domain accuracy (D-acc). The human evaluation results (Table 4.5) show a strong preference of DAST over DAST-C as well as ControlGen in terms of style control, content preservation,

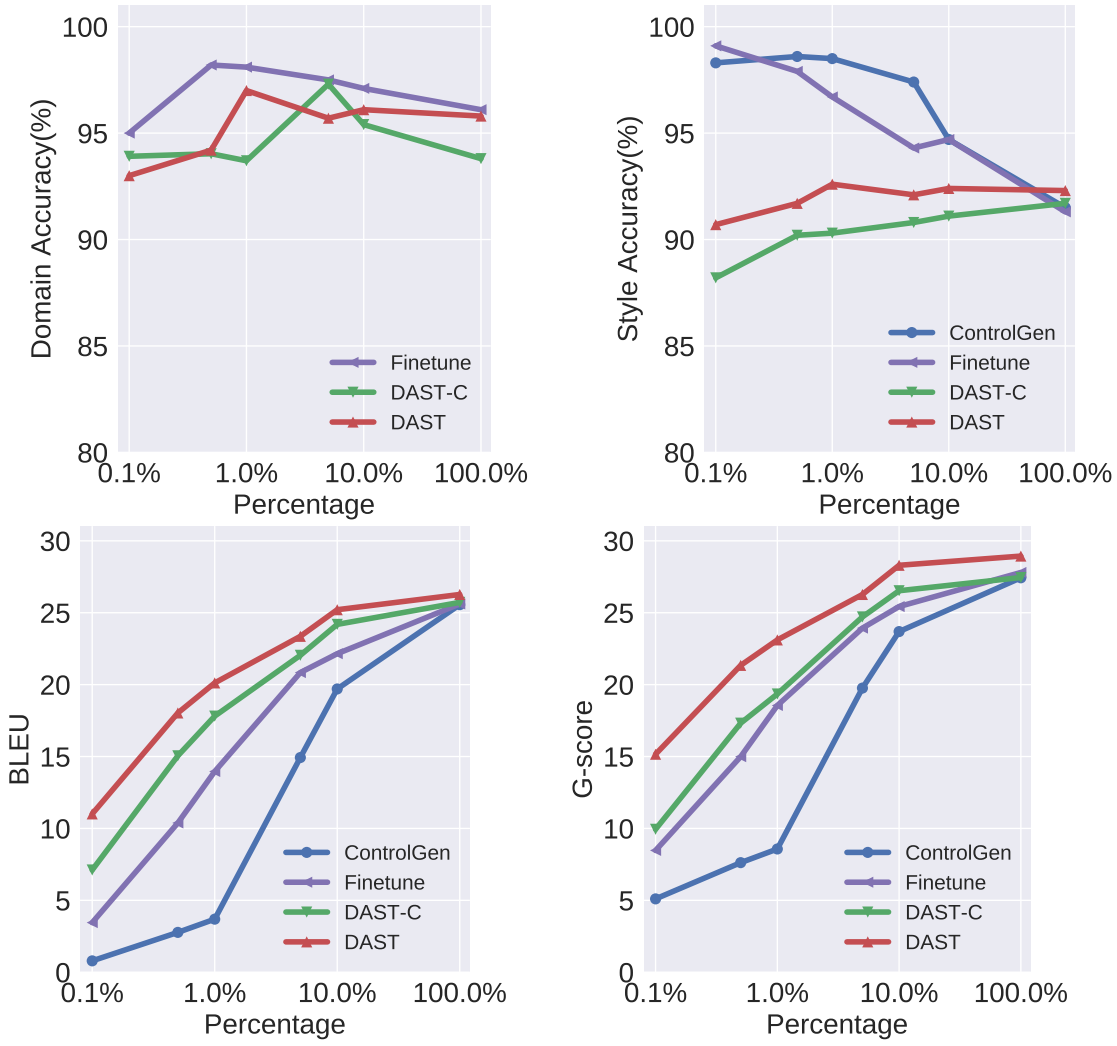


Figure 4.2: Results on Yelp test set in terms of different percentage of target domain data. $0.1\% \approx 400$ samples.

Yelp (1% target data)									
Style Control					**Content Preservation**				
Our Model		Neutral	Comparison		Our Model		Neutral	Comparison	
DAST	**56.2**%	30.5%	13.3%	ControlGen	DAST	**47.0**%	48.4%	4.6%	ControlGen
DAST	**40.5**%	42.3%	17.2%	DAST-C	DAST	**22.4**%	65.7%	11.9%	DAST-C
DAST	17.9%	18.5%	**63.6**%	Human	DAST	17.7%	47.4%	**34.9**%	Human
Fluency					**Overall Quality**				
Our Model		Neutral	Comparison		Our Model		Neutral	Comparison	
DAST	**47.1**%	40.8%	12.0%	ControlGen	DAST	**81.1**%	14.0%	4.9%	ControlGen
DAST	**29.1**%	55.8%	15.1%	DAST-C	DAST	**31.4**%	43.0%	25.6%	DAST-C
DAST	10.1%	30.4%	**59.5**%	Human	DAST	16.9%	23.9%	**59.2**%	Human
Enron									
Style Control					**Content Preservation**				
Our Model		Neutral	Comparison		Our Model		Neutral	Comparison	
DAST	**74.2**%	19.8%	6%	ControlGen	DAST	**80.8**%	14.8%	4.4%	ControlGen
DAST	**28.4**%	50.2%	21.4%	DAST-C	DAST	**26.8**%	48.8%	24.4%	DAST-C
DAST	17.6%	30.5%	**51.9**%	Human	DAST	15.3%	36.9%	**47.8**%	Human
Fluency					**Overall Quality**				
Our Model		Neutral	Comparison		Our Model		Neutral	Comparison	
DAST	**73.8**%	20.6%	5.6%	ControlGen	DAST	**52.7**%	35.3%	12.0%	ControlGen
DAST	**26.9**%	51.6%	21.5%	DAST-C	DAST	**34.0**%	48.4%	17.6%	DAST-C
DAST	11.6%	36.5%	**51.9**%	Human	DAST	12.0%	17.8%	**68.0**%	Human

Table 4.5: Results of **Human Evaluation** for style control, content preservation, fluency and overall quality showing preferences (%) for DAST model vis-a-vis baseline or other comparison systems.

Yahoo Sentiment Transfer			
Model	**D-acc**	**S-acc**	**BLEU**
ControlGen	-	99.1	9.7
Finetune	**97.8**	98.8	31.4
DAST-C	90.7	98.8	35.9
DAST	90.8	**99.2**	**39.2**

Table 4.6: Results on Yahoo sentiment transfer task.

Model	**D-acc**	**S-acc**	***h*BLEU**	**G-score**
DAST	**97.0**	**92.6**	**20.1**	**23.1**
w/o domain-specific attributes	83.9	90.9	20.0	22.7
w/o domain-specific classifiers	91.4	83.8	19.0	20.8
w/o both	73.8	80.6	18.7	19.9

Table 4.7: Ablation study on Yelp (1%) dataset with help from IMDB dataset. The results are evaluated on Yelp test set.

fluency and overall quality of generated samples. The samples of Yelp sentiment transfer are shown in Table 4.4.

Finally, we evaluate our models on Yahoo sentiment transfer task. As can be seen in Table 4.6, both DAST and DAST-C achieve successful style transfer even if the target data is formed as questions which have a large discrepancy with the source IMDB domain.

Limiting the Target Domain Data. We further test the limit of our model by using as few target domain data as possible. Figure 4.2 shows the quantitative results with different percentages of target domain training data. When the target domain data is insufficient,

Training Setup	D-acc	S-acc	*h*BLEU	G-score
IMDB+Yelp	97.0	92.6	20.1	**23.1**
Finetune	**98.1**	96.7	13.9	18.5
IMDB	62.8	59.3	**21.4**	12.2
Yelp	96.8	**98.5**	3.7	8.6

Table 4.8: Results of different training setups when using Yelp (1%) and IMDB datasets. The results are evaluated on Yelp test set.

Model	Source	# Samples	D-acc	S-acc	BLEU
DAST-C	IMDB	572K	96.9	90.3	17.8
	Yahoo	900K	90.3	91.3	19.6
	GYAFC	206K	93.5	92.9	16.1
DAST	IMDB	334K	97.0	92.6	20.1
	TripAdvisor	572K	86.2	91.4	18.4

Table 4.9: Performance on the Yelp (1% data) dataset with help of different source domain data. The results are evaluated on Yelp test set.

especially less than 10%, the content preservation ability of the baseline (trained with target data only) has degenerated rapidly despite a relatively high style transfer accuracy. This is less than desirable because by retrieving sentences with the target style a transferred sentence can easily exhibit the correct style while retaining barely any content similar to the input. Finetune improves content preservation but still suffers the same problem with less target data. Note that DAST-C is not comparable to Finetune as the former does not use the style information in the source domain.

Both DAST models bring substantial improvements to content preservation, and can

Enron Formality Transfer			
Model	**D-acc**	**S-acc**	***h*BLEU**
ControlGen	-	81.2	4.7
Finetune	**91.3**	81.6	14.7
DAST-C	87.6	89.2	15.5
DAST	88.4	**91.6**	**16.4**

Table 4.10: Results on Enron formality transfer tasks.

still successfully manipulate the styles, resulting in consistently higher G-scores. This is presumably because our models adapt the content information as well as the style information from the source domain to consistently sustain the style transfer on the target domain. By learning both generic and domain-specific stylized information, DAST outperforms DAST-C in terms of content preservation and style control. Even with 0.1% target domain data (400 samples), DAST was able to attain a reasonable degree of text style transfer, whereas the model trained on the target data generated entirely nonsensical sentences. Meanwhile, DAST succeeded in transferring the sentences in a domain-aware manner, achieving consistently high domain accuracy.

Ablation Study. To investigate the effect of individual components and training setup on the overall performance, we conduct an ablation study in Table 4.7. The domain vectors enable the model to transfer sentences in a domain-aware manner, and thus give the largest boost on domain accuracy. Without domain-specific style classifiers, the model mixes the style information on both domains, resulting in worse style control and content preservation. Additionally, simply increasing the number of training examples (i.e., the row "w/o both") improves content preserving, while introducing a data distribution discrepancy between the training (Yelp+IMDB) and test data (Yelp), as evidenced by the lower S-acc and D-acc

Enron (informal-to-formal)	
Input	ya 'll need to come visit us in austin .
ControlGen	**could we need to look on saturday in enpower .**
Finetune	*you will* need to go **in bed with him** .
DAST-C	*you will* need to visit **town** .
DAST	*yes , you will* need to visit us in austin .
Human	all of you should come visit us in austin .
Enron (formal-to-informal)	
Input	are n't you suppose to be teaching some kids or something ?
ControlGen	*are you not* supposed to be **disloyal** some kids or something ?
Finetune	*are you not* **to be able to be some man** or something ?
DAST-C	are not you supposed to be teaching some kids or something ?
DAST	*are you not* supposed to be teaching some *children* or something ?
Human	are you not supposed to be instructing children ?

Table 4.11: Transferred sentences on Enron dataset, where *italic red* denotes successful style transfers, **bold blue** denotes content losses, and sans serif orange denotes grammar errors. Better looked in color.

scores.

In terms of the training setup (Table 4.8), the source domain IMDB mostly helps content preservation, while accurate style information is mainly learned from the target domain Yelp. Finetune gives higher S-acc and D-acc and lower hBLEU due to catastrophic forgetting. Our proposed DAST successfully exploits the source domain data, and thus yields balanced results on style and domain control, while maintaining content preservation.

To investigate the effectiveness of the source domain data, we evaluate our proposed models on different source domains that have unknown styles or the same styles as Yelp. Results are included in Table 4.9. It can be seen that the proposed models can robustly

achieve favorable style transfer with help of different source domain data. Since DAST-C model mainly learns the generic content information by modeling the large corpus on the source domain, the number of source training data significantly affects the performance, especially on content preservation (BLEU). On the other hand, since DAST also adapts generic style information, the source domain that has closer sentiment information (IMDB) provides more benefit to the target domain (Yelp) than the TripAdvisor dataset does.

Non-parallel Style Transfer with Parallel Source Data. Finally, to verify the versatility of our proposed models in different scenarios, we investigate another domain adaptation setting, where the source domain data (GYAFC) is parallel but the target domain data (Enron) is non-parallel. Since parallel data is available in the source domain, we are able to simply add a sequence-to-sequence loss $L^{\mathcal{S}}_{s2s}$ on source domain data in Eqn. (4.6) and Eqn. (4.9) to help the target domain without parallel data. The training objectives can be written as: $L^{\mathcal{T}}_{ae} + L^{\mathcal{T}}_{style} + L^{\mathcal{S}}_{ae} + L^{\mathcal{S}}_{s2s}$ and $L^{\mathcal{S},\mathcal{T}}_{ae} + L^{\mathcal{S},\mathcal{T}}_{style} + L^{\mathcal{S}}_{s2s}$, respectively. Results are summarized in Table 4.10. DAST outperforms other methods on both style control and content preservation while keeping the transferred sentences with target-specific characteristics (D-acc). A strong human preference for DAST can be observed in Table 4.5 when compared to the baselines. Qualitative samples are provided in Table 4.11.

4.6 Conclusion

We present two simple yet effective domain adaptive text style transfer models that leverage massively available data from other domains to facilitate the transfer task in the target domain. The proposed models achieve better content preservation with the generic information learned from the source domain and simultaneously distinguish the domain-specific information, which enables the models to transfer text in a domain-adaptive manner. Extensive experiments demonstrate the robustness and applicability on various scenarios where the target data is limited.

Chapter 5

CONTEXTUALIZED PERTURBATION FOR TEXTUAL ADVERSARIAL ATTACK

5.1 Introduction

Adversarial example generation for natural language processing (NLP) tasks aim to perturb input text to trigger errors in machine learning models, while keeping the output close to the original. Besides exposing system vulnerabilities and helping improve their robustness and security [169, 138, 14, 57, *inter alia*], adversarial examples are also used to analyze and interpret the models' decisions [56, 116].

Generating adversarial examples for NLP tasks can be challenging, in part due to the discrete nature of natural language text. Recent efforts have explored heuristic rules, such as replacing tokens with their synonyms [118, 77, 2, 111, 58, *inter alia*]. Despite some empirical success, rule-based methods are agnostic to context, limiting their ability to produce natural, fluent, and grammatical outputs [142, 94, 66, *inter alia*].

This work presents CLARE, a **C**ontextua**L**ized **A**dversa**R**ial **E**xample generation model for text. CLARE perturbs the input with a mask-then-infill procedure: it first detects the vulnerabilities of a model and deploys masks to the inputs to indicate missing text, then fills in an alternative token using a pretrained masked language model (e.g., RoBERTa; [85]). CLARE features three contextualized perturbing actions: *Replace*, *Insert* and *Merge*, which respectively replace a token, insert a new token, and merge a bigram (Figure 5.1). As a result, it can generate outputs of varied lengths, in contrast to token replacement based methods that only produce outputs of the same lengths as the inputs [2, 111, 58]. Further, CLARE searches over a wider range of attack strategies, and is thus able to attack the victim model more efficiently with fewer edits. Building on a masked language model, CLARE maximally

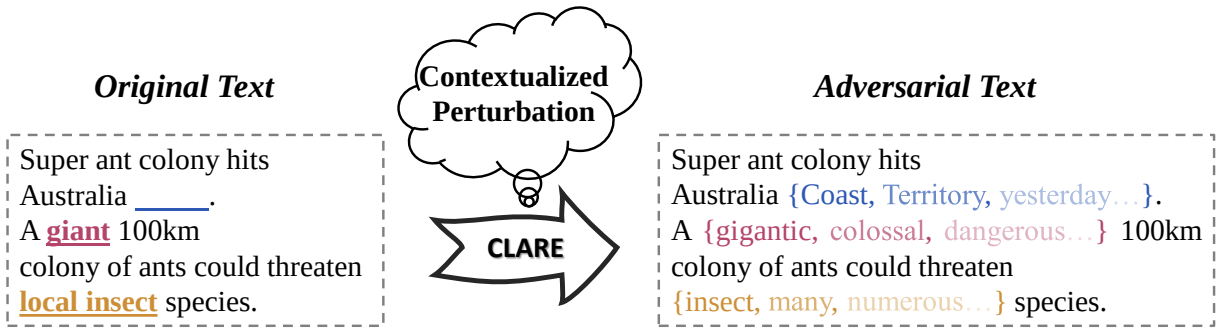


Figure 5.1: Illustration of CLARE. Through a mask-then-infill procedure, the model generates the adversarial text with three contextualized perturbations: ***Replace***, ***Insert*** and ***Merge***. A mask is indicated by "__". The degree of fade corresponds to the (decreasing) priority of the infill tokens.

preserves textual similarity, fluency, and grammaticality of the outputs.

We evaluate CLARE on text classification, natural language inference, and sentence paraphrase tasks, by attacking finetuned BERT models [21]. Extensive experiments and human evaluation show that CLARE outperforms baselines in terms of attack success rate, textual similarity, fluency, and grammaticality, and strikes a better balance between attack success rate and preserving input-output similarity. Our analysis further suggests that the CLARE can be used to improve the robustness of the downstream models, and improve their accuracy when the available training data is limited. We will open-source our code and models upon publication.

5.2 Related Work

Textual adversarial attack. An increasing amount of effort is being devoted to generating better textual adversarial examples with various attack models. Character-based models [77, 25, 71, 31, *inter alia*] use misspellings to attack the victim systems; however,

these attacks can often be defended by a spell checker [103, 136, 171, 60]. Many sentence-level models [51, 140, 173, *inter alia*] have been developed to introduce more sophisticated token/phrase perturbations. These, however, generally have difficulty maintaining semantic similarity with original inputs [159]. Recent word-level models explore synonym substitution rules to enhance semantic meaning preservation [2, 58, 111, 157, 156, *inter alia*]. Our work differs in that CLARE uses three contextualized perturbations that can produce more fluent and grammatical outputs.

Text generation with BERT. Generation with masked language models has been widely studied in various natural language tasks, ranging from lexical substitution [148, 170, 104, 149, *inter alia*] to non-autoregressive generation [41, 70, 33, 88, 127, 112, 168, *inter alia*]. However, little work has explored using these models to generate adversarial examples for text.

5.3 Proposed Model

At a high level, CLARE applies a sequence of contextualized perturbation actions to the input. Each can be seen as a *local* mask-then-infill procedure: it first applies a mask to the input around a given position, and then fills it in using a pretrained masked language model (§5.3.1). To produce the output, CLARE scores and descendingly ranks the actions, which are then iteratively applied to the input (§5.3.2). We begin with a brief background review and laying out of necessary notation.

Background. Adversarial example generation centers around a **victim** model f, which we assume is a text classifier. We focus on the black-box setting, allowing access to f's outputs but *not* its configurations such as parameters. Given an input sequence $\mathbf{x} = x_1 x_2 \dots x_n$ and its label y, assume $f(\mathbf{x}) = y$, an **adversarial example** $\mathbf{x}'$ is supposed to modify $\mathbf{x}$ to trigger an error in the victim model: $f(\mathbf{x}') \neq f(\mathbf{x})$. At the same time, textual modifications should be minimal, such that $\mathbf{x}'$ is close to $\mathbf{x}$ and the human predictions on $\mathbf{x}'$ stay the same.

In computer vision applications, minor perturbations to continuous pixels can be barely perceptible to humans, thus it can be hard for one to distinguish $\mathbf{x}$ and $\mathbf{x}'$ [38]. It is not the case for text, however, since changes to the discrete tokens are more likely to be noticed by humans.

This is achieved by requiring the similarity between $\mathbf{x}'$ and $\mathbf{x}$ to be larger than a threshold: $\text{sim}(\mathbf{x}', \mathbf{x}) > \ell$. A common choice of $\text{sim}(\cdot, \cdot)$ is to encode sentences using neural networks, and calculate their cosine similarity in the embedding space [58].

5.3.1 Masking and Contextualized Infilling

At a given position of the input sequence, CLARE can execute three perturbation actions: *Replace*, *Insert*, and *Merge*, which we introduce in this section. These apply masks at the given position with different strategies, and then fill in the missing text based on the unmasked context.

***Replace*:** A *Replace* action substitutes the token at a given position i with an alternative (e.g., changing "*fantastic*" to "*amazing*" in "The movie is *fantastic*."). It first replaces x_i with a mask, and then selects a token z from a candidate set $\mathcal{Z}$ to fill in:

$$\widetilde{\mathbf{x}} = x_1 \dots x_{i-1} \ [\textsc{Mask}] \ x_{i+1} \dots x_n,$$
$$\text{replace}\,(\mathbf{x}, i) = x_1 \dots x_{i-1} \ z \ x_{i+1} \dots x_n.$$

Note that $\widetilde{\mathbf{x}}$ and $\widetilde{\mathbf{x}}_z$ depend on i, and $\mathcal{Z}$ depends on $\mathbf{x}$ and i. For clarity, we suppress such dependence and denote $\text{replace}\,(\mathbf{x}, i)$ by $\widetilde{\mathbf{x}}_z$.

To produce an adversarial example,

- z should fit into the unmasked context;
- $\widetilde{\mathbf{x}}_z$ should be similar to $\mathbf{x}$;
- $\widetilde{\mathbf{x}}_z$ should trigger an error in f.

These can be achieved by selecting a z such that

- z receives a high probability from a masked language model: $p_{\text{MLM}}(z \mid \widetilde{\mathbf{x}}) > k$;

- $\widetilde{\mathbf{x}}_z$ is similar to $\mathbf{x}$: $\text{sim}(\mathbf{x}, \widetilde{\mathbf{x}}_z) > \ell$;
- f predicts low probability for the gold label given $\widetilde{\mathbf{x}}_z$, i.e., $p_f(y \mid \widetilde{\mathbf{x}}_z)$ is small.

p_{MLM} denotes a pretrained masked language model (e.g., RoBERTa; [85]). Using higher k, ℓ thresholds produces outputs that are more fluent and closer to the original. However, this can undermine the success rate of the attack. We choose k, ℓ to trade-off between these two aspects. k and ℓ are empirically set as 5×10^{-3} and 0.7, respectively. This also reduces the computation overhead: in our experiments $|\mathcal{Z}|$ is 42 on average, much smaller than the vocabulary size ($|\mathcal{V}| = 50,265$).

The first two requirements can be met by the construction of the candidate set: $\mathcal{Z} =$

$$\{z' \in \mathcal{V} \mid p_{\text{MLM}}(z' \mid \widetilde{\mathbf{x}}) > k, \text{sim}(\mathbf{x}, \widetilde{\mathbf{x}}_{z'}) > \ell\}.$$

$\mathcal{V}$ is the vocabulary of the masked language model. To meet the third, we select from $\mathcal{Z}$ the token that, if filled in, will cause most "confusion" to f:

$$z = \arg\min_{z' \in \mathcal{Z}} p_f(y \mid \widetilde{\mathbf{x}}_{z'}).$$

The *Insert* and *Merge* actions differ from *Replace* in terms of masking strategies. The alternative token z is selected analogously to that in a *Replace* action.

***Insert*:** This aims to add extra information to the input (e.g., changing "I recommend ..." to "I *highly* recommend ..."). It inserts a mask after x_i and then fills it. Slightly overloading the notations,

$$\widetilde{\mathbf{x}} = x_1 \dots x_i \ [\text{MASK}] \ x_{i+1} \dots x_n,$$
$$\text{insert}\,(\mathbf{x}, i) = x_1 \dots x_i \ z \ x_{i+1} \dots x_n.$$

This increases the sequence length by 1.

***Merge*:** This masks out a bigram $x_i x_{i+1}$ with *a single* mask and then fills it, reducing the sequence length by 1:

$$\widetilde{\mathbf{x}} = x_1 \dots x_{i-1} \text{ [MASK] } x_{i+2} \dots x_n,$$
$$\text{merge}\,(\mathbf{x}, i) = x_1 \dots x_{i-1}\ z\ x_{i+2} \dots x_n.$$

z can be the same as one of the masked tokens (e.g., masking out "New York" and then filling in"York"). This can be seen as deleting a token from the input.

For *Insert* and *Merge*, z is chosen in the same manner as replace action.

Note that each candidate is represented by a subword unit before de-tokenization. Besides x_i, candidates constructed by multiple subwords are also not included in $\mathcal{Z}$. A perturbation will not be considered if its candidate token set is empty.

In sum, at each position i of an input sequence, CLARE first: (1) replaces x_i with a mask; (2) or inserts a mask after x_i; (3) or merges $x_i x_{i+1}$ into a mask. Then a set of candidate tokens is constructed with a masked language model and a textual similarity function; the token minimizing the gold label's probability is chosen as the alternative token.

CLARE first constructs the local actions for all positions in parallel, i.e., the actions at position i do not affect those at other positions. Then, to produce the adversarial example, CLARE gathers the local actions and selects an order to execute them.

5.3.2 Sequentially Applying the Perturbations

Given an input pair $(\mathbf{x}, y)$, let n denote the length of $\mathbf{x}$. CLARE chooses from $3n$ actions to produce the output: 3 actions for each position, assuming the candidate token sets are not empty. We aim to generate an adversarial example with minimum modifications to the input. To achieve this, we iteratively apply the actions, and first select those minimizing the probability of outputting the gold label y from f.

Each action is associated with a score, measuring how likely it can "confuse" f: denote by $a(\mathbf{x})$ the output of applying action a to $\mathbf{x}$. The score is then the negative probability of

predicting the gold label from f, using $a(\mathbf{x})$ as the input:

$$s_{(\mathbf{x},y)}(a) = -p_f\big(y \mid a(\mathbf{x})\big).$$

Only one of the three actions can be applied at each position, and we select the one with the highest score. This constraint aims to avoid multiple modifications around the same position, e.g., merging "New York" into "Seattle" and then replacing it with "Boston". Note that multiple actions at the same position can be replaced by one. In preliminary experiments, we found that constraining one action per position yields better performance in terms of fluency and grammaticality.

Actions are iteratively applied to the input, until an adversarial example is found or a limit of actions T is reached. If no adversarial example is found after applying T actions, it counts as a failed attack. Each step selects the highest-scoring action from the remaining ones. Algorithm 1 summarizes the above procedure. Note that *Insert* and *Merge* actions change the text length. When any of them is applied, we accordingly change the text indices of affected actions remaining in $\mathcal{A}$.

Discussion. A key technique of CLARE is the local mask-then-infill perturbation. This comes with several advantages. First, it allows attacking *any* position of the input sequence, whereas existing synonym replacement approaches can generally only attack tokens in a predefined vocabulary [2, 58, 111, *inter alia*]. Second, as we will show in the experiments (§5.4.3), contextualized infilling produces more fluent and grammatical outputs compared to the context-agnostic counterparts, especially when using masked language models trained on large-scale data. In addition, by using *Merge* and *Insert* actions, CLARE can produce adversarial examples whose lengths are different from the inputs.

Generating adversarial examples with masked language models is also explored by a concurrent work [76]. Their method is similar to a CLARE model except that it only uses the *Replace* action. As shown in our ablation study (§5.5.1), using all three actions helps CLARE achieve a better attack performance.

Algorithm 1 Adversarial Attack by CLARE

Input: Text-label pair $(\mathbf{x}, y)$; Victim model f
Output: An adversarial example
Initialization: $\mathbf{x}^{(0)} = \mathbf{x}$
$\mathcal{A} \leftarrow \varnothing$
for $1 \leq i \leq |\mathbf{x}|$ **do**
    $a \leftarrow$ highest-scoring action from $\{$
        $\text{replace}(\mathbf{x}, i), \text{insert}(\mathbf{x}, i), \text{merge}(\mathbf{x}, i)\}$
    $\mathcal{A} \leftarrow \mathcal{A} \bigcup \{a\}$
end for
for $1 \leq t \leq T$ **do**
    $a \leftarrow$ highest-scoring action from $\mathcal{A}$
    $\mathcal{A} \leftarrow \mathcal{A} \setminus \{a\}$
    $\mathbf{x}^{(t)} \leftarrow$ Apply a on $\mathbf{x}^{(t-1)}$
    if $f(\mathbf{x}^{(t)}) \neq y$ **then return** $\mathbf{x}^{(t)}$
    end if
end for
return NONE

5.4 Experiments

We evaluate CLARE on text classification, natural language inference, and sentence paraphrase tasks. We begin by describing the implementation details of CLARE and the baselines (§5.4.1). §5.4.2 introduces the datasets we experiment with and the evaluation metrics; the results are summarized in §5.4.3. All experiments are conducted on one Nvidia GTX 1080Ti GPU.

5.4.1 Setup

- We experiment with a distilled version of RoBERTa (RoBERTa$_{\text{distill}}$; [119]) as the masked language model for contextualized infilling. We also compare to base sized RoBERTa (RoBERTa$_{\text{base}}$; [85]) and base sized BERT (BERT$_{\text{base}}$; [21]) in the ablation study (§5.5.1).
- The similarity function builds on the universal sentence encoder (USE; [10]).
- The victim model is an MLP classifier on top of BERT$_{\text{base}}$. It takes as input the first token's contextualized representation. We finetune BERT when training the victim model.
- *Merge* perturbation can only merge noun phrases, extracted with the NLTK toolkit.[1] We find that this helps produce more grammatical outputs.

All pretrained models and victim models based on RoBERTa and BERT$_{\text{base}}$ are implemented with Hugging Face transformers[2] [147] based on PyTorch [99]. RoBERTa$_{\text{distill}}$, RoBERTa$_{\text{base}}$ and uncase BERT$_{\text{base}}$ models have 82M, 125M and 110M parameters, respectively. We use RoBERTa$_{\text{distill}}$ as our main backbone for fast inference purpose.

Baselines. We compare CLARE with recent state-of-the-art word-level black-box adversarial attack models, including:

- **PWWS**: a recent model by [111]. Based on word saliency [72], it greedily replaces tokens with their synonyms from WordNet [91].
- **TextFooler**: a state-of-the-art model by [58]. This replaces tokens with their synonyms derived from counter-fitting word embeddings [95], and uses the same text similarity function as our work.
- **TextFooler+LM**: an improved variant of TextFooler we implemented based on [2] and [14]. This inherits token replacement from TextFooler, but uses an additional

[1] `https://www.nltk.org/`

[2] `https://github.com/huggingface/transformers`

Dataset	Avg. Length	# Classes	Train	Test	Acc
Yelp	130	2	560K	38K	95.9%
AG News	46	4	120K	7.6K	95.0%
DBpedia	55	14	560K	70K	99.3%
MNLI[5]	23/11	3	392K	9.8K	84.3%
QNLI	11/31	2	105K	5.4K	91.4%
MRPC	23/23	2	3.6K	1.7K	81.4%

Table 5.1: Some statistics of datasets. The last column indicates the victim model's accuracy on the original test set *without* adversarial attack.

small sized GPT-2 language model [107] to filter out those candidate tokens that do not fit in the context with calculated perplexity.

PWWS[3] and TextFooler[4] are built with their open source implementation provided by the authors. In the implementation of TextFooler+LM, we use small sized GPT-2 language model [107] to further select those candidate tokens that have top 20% perplexity in the candidate token set.

The similarity function sim builds on the universal sentence encoder (USE; [10]) to measure a *local* similarity at the perturbation position with window size 15 between the original input and its adversary.

In the adversarial training (§5.5.2), the small TextCNN victim model [63] has 128 embedding size and 100 filters for $3, 4, 5$ window size with 0.5 dropout, resulting in 7M parameters.

[3] `https://github.com/JHL-HUST/PWWS/`

[4] `https://github.com/jind11/TextFooler`

[5] We only examine the performance on the matched set, since the mismatched set is easier to attack.

	Yelp (PPL = 51.5)					**AG News** (PPL = 62.8)				
Model	**A-rate↑**	**Mod↓**	**PPL↓**	**GErr↓**	**Sim↑**	**A-rate↑**	**Mod↓**	**PPL↓**	**GErr↓**	**Sim↑**
PWWS	35.3	**8.2**	98.8	0.33	0.64	14.2	7.9	114.8	0.56	0.71
TextFooler	77.0	16.6	163.3	1.23	0.70	56.1	23.3	331.3	1.43	0.69
+ LM	34.0	17.4	90.0	1.21	0.73	23.1	21.9	144.6	1.07	0.74
CLARE	**79.1**	10.3	**83.5**	**0.25**	**0.78**	**65.3**	**5.9**	**82.9**	**0.15**	**0.76**
	MNLI (PPL = 60.9)					**QNLI** (PPL = 46.0)				
Model	**A-rate↑**	**Mod↓**	**PPL↓**	**GErr↓**	**Sim↑**	**A-rate↑**	**Mod↓**	**PPL↓**	**GErr↓**	**Sim↑**
PWWS	16.6	**6.4**	101.3	0.30	0.70	8.8	**8.0**	88.4	0.32	0.71
TextFooler	59.8	13.8	161.5	0.63	0.73	57.8	16.9	164.4	0.62	0.72
+ LM	32.3	12.4	91.9	0.50	0.77	29.2	17.3	85.0	0.42	0.75
CLARE	**88.1**	7.5	**82.7**	**0.02**	**0.82**	**83.8**	11.8	**76.7**	**0.01**	**0.78**
	DBpedia (PPL = 37.3)					**MRPC** (PPL = 42.9)				
Model	**A-rate↑**	**Mod↓**	**PPL↓**	**GErr↓**	**Sim↑**	**A-rate↑**	**Mod↓**	**PPL↓**	**GErr↓**	**Sim↑**
PWWS	7.6	8.3	57.6	0.54	0.68	5.8	**6.5**	82.6	0.31	0.68
TextFooler	56.2	24.9	182.5	1.88	0.68	24.5	10.6	118.8	0.35	0.75
+ LM	20.1	22.4	84.0	1.22	0.70	12.9	9.5	71.0	0.29	0.79
CLARE	**65.8**	**7.02**	**53.3**	**-0.03**	**0.73**	**34.8**	9.1	**69.5**	**0.02**	**0.83**

Table 5.2: Adversarial example generation performance in attack success rate (A-rate), modification rate (Mod), perplexity (PPL), number of increased grammar errors (GErr), and textual similarity (Sim). The perplexity of the original inputs is indicated in parentheses for each dataset. Bold font indicates the best performance for each metric.

5.4.2 Datasets and Evaluation

Datasets. We evaluate CLARE with the following datasets:

- **Yelp Reviews** [161]: a binary sentiment classification dataset based on restaurant reviews.
- **AG News** [161]: a collection of news articles with four categories: *World*, *Sports*, *Business* and *Science & Technology*.
- **DBpedia** [161]: a dataset of structured texts extracted from Wikipedia with 14 non-overlapping classes.
- **MNLI** [145]: a natural language inference dataset. Each instance consists of a premise-hypothesis pair, and the model is supposed to determine the relation between them from a label set of *entailment*, *neutral*, and *contradiction*. It covers text from a variety of domains.
- **QNLI** [139]: a binary classification dataset converted from the Stanford question answering dataset [108]. The task is to determine whether the context contains the answer to a question. It is mainly based on English Wikipedia articles.
- **MRPC** [23]: is a corpus of sentence pairs automatically extracted from online news sources, with human annotations for whether the sentences in the pair are semantically equivalent.

While **AG News**, **Yelp Reviews** and **DBpedia** datasets are from [161], **MNLI**, **QNLI** and **MRPC** datasets are obtained from GLUE benchmark [139]. Table 5.1 summarizes some statistics of the datasets. Following previous practice [2], we tune CLARE on training data, and evaluate with 1,000 randomly sampled test instances of lengths ≤ 100. When processing the data, we keep all punctuation in texts for both victim model training and attacking. Since GLUE benchmark [139] does not provide the label for test set, we instead use its dev set as the the test set for the included datasets (MNLI, QNLI, MRPC) in the evaluation. For the sentence-pair tasks (MNLI, QNLI, MRPC), we attack the longer one excluding the tokens appearing in both sentences. This is because inference tasks usually require entailed

data to have the same keywords, e.g., numbers, name entities, etc.

Evaluation metrics. We follow previous works [58, 156], and evaluate the models with the following automatic metrics:

- **Attack success rate (A-rate)**: the percentage of adversarial examples that can successfully attack the victim model.
- **Modification rate (Mod)**: the percentage of modified tokens. Each *Replace* or *Insert* action accounts for one token modified; a *Merge* action is considered modifying one token if one of the two merged tokens is kept (e.g., merging bigram ab into a), and two otherwise (e.g., merging bigram ab into c).
- **Perplexity (PPL)**: a metric used to evaluate the *fluency* of adversaries [61, 156]. The perplexity is calculated using small sized GPT-2 with a 50K-sized vocabulary [107].
- **Grammar error (GErr)**: the number of increased grammatical errors in the successful adversarial example, compared to the original text. Following [156, 94], we calculate this by the LanguageTool [96].[6]
- **Textual similarity (Sim)**: the similarity between the input and its adversary. Following [58, 94], we calculate this using the universal sentence encoder (USE; [10]).

The last four metrics are averaged across those adversarial examples that successfully attack the victim model.

The evaluation metric **Sim** uses USE to calculate a *global* similarity between two texts. This procedure is typically following [58]. We mostly rely on human evaluation (§5.4.3) to conclude the significant advantage of preserving textual similarity on CLARE compared with TextFooler.

5.4.3 Results

Table 5.2 summarizes the results. Although PWWS achieves the best modification rate on 3 out of the 4 datasets, it *underperforms* CLARE in terms of other metrics. With a very

[6] `https://www.languagetool.org/`

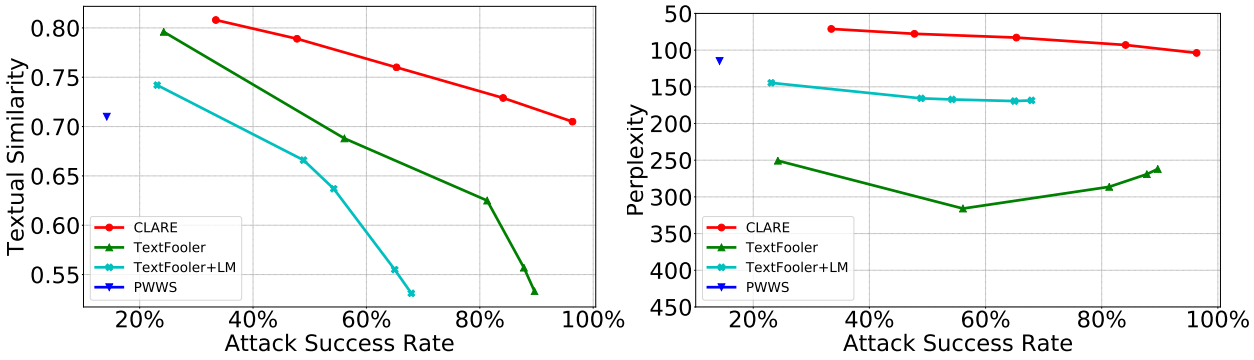


Figure 5.2: Attack success rate / Textual similarity trade-off curves (both higher the better) and Attack Success Rate (*higher is better*) / Perplexity (*lower is better*) trade-off curves. The larger area under the curve indicates the better trade-off between the two.

limited set of synonym candidates from WordNet, PWWS fails to attack a BERT model on most of inputs. Using word embeddings to find synonyms, TextFooler achieves a higher success rate, but tends to produce less grammatical and less natural outputs. Equipped with a language model, TextFooler+LM does better in terms of perplexity, yet this brings little grammaticality improvement and comes at a cost to attack success rate. With contextualized perturbations, CLARE achieves the best performance on attack success rate, perplexity, grammaticality and similarity. For AG News, CLARE outperforms TextFooler by 9% on success rate and by a huge 245 on perplexity, and cuts average number of grammatical errors by 1.3. We observe similar trends on other datasets.

Figure 5.2 compares trade-off curves between attack success rate and textual similarity. For each model, we tune the thresholds for constructing the candidate token sets, and plot textual similarity against attack success rate. CLARE strikes the best balance, showing a clear advantage in achieving a success rate with least similarity drop. We observe similar trends for success rate and perplexity trade off.

Human evaluation. We further conduct human evaluation on the AG News dataset. We randomly sampled 300 sentences from the test set combining the corresponding adversarial

examples from CLARE and TextFooler (We only consider sentences can be attacked by both models). In order to make the task less abstract, we pair the adversarial examples by the two models, and present them to the participants along with the original input and its gold label. We ask them which one they prefer in terms of (1) having more similar a meaning to the original input (similarity), and (2) being more fluent and grammatical (fluency and grammaticality). We also provide them with a neutral option, when the participants consider the two indistinguishable. Additionally, we ask the participants to annotate the adversarial examples, and compare their annotations against the gold labels (label consistency). Higher label consistency indicates the model is better at causing the victim model to make errors while preserving human predictions.

Each pair of system outputs was randomly presented to 5 crowd-sourced judges, who indicated their preference for similarity, fluency, and grammaticality using. To minimize the impact of spamming, we employed the top-ranked 30% of U.S. workers provided by the crowd-sourcing service. Detailed task descriptions and examples were also provided to guide the judges. We calculate p-value based on 95% confidence intervals by using 10K paired bootstrap replications, implemented using the R Boot statistical package.

As shown in Table 5.3, CLARE has a significant advantage over TextFooler: in terms of similarity 56% responses prefer CLARE, while 16% prefer TextFooler. The trend is similar for fluency & grammaticality (42% vs. 9%). On label consistency, CLARE slightly underperforms TextFooler at 68% with a 95% condidence interval (CI) $(66\%, 70\%)$, versus 70% with a 95% CI $(68\%, 73\%)$. We attribute this to an inherent overlap of some categories in the AG News dataset, e.g., *Science & Technology and Business*, as evidenced by a 71% label consistency for original inputs.

Closing this section, Table 5.8 and Table 5.9 compare the adversarial examples generated by TextFooler and CLARE.

Metric	CLARE	Neutral	TextFooler
Similarity	$56.1_{\pm 2.5}$	28.1	$15.8_{\pm 2.1}$
Fluency&Grammaticality	$42.5_{\pm 2.5}$	48.6	$8.9_{\pm 1.5}$
Label Consistency	$68.0_{\pm 2.4}$	-	$70.1_{\pm 2.5}$

Table 5.3: Human evaluation performance in percentage on the AG News dataset. ± indicates confidence intervals with a 95% confidence level.

5.5 Analysis

This section first conducts an ablation study (§5.5.1). We then explore CLARE's potential to be used to improve downstream models' robustness and accuracy in §5.5.2. In §5.5.3, we empirically observe that CLARE tends to attack noun and noun phrases.

5.5.1 Ablation Study

We ablate each component of CLARE to study its effectiveness. We evaluate on the 1,000 randomly selected AG news instances (§5.4.2). The results are summarized in Table 5.4.

We first investigate the performance of three perturbations when applied individually. Among three editing strategies, using InsertOnly achieves the best performance, with ReplaceOnly coming in a close second. MergeOnly underperforms the other two, partly due to that the attacks are restricted to bigram noun phrases (§5.4.1). Combining all three perturbations, CLARE achieves the best performance with the least modifications.

To examine the effect of contextualized infilling, we compare ReplaceOnly against TextFooler, a context-agnostic model based on token replacement. ReplaceOnly outperforms TextFooler across the board, suggesting that contextualized infilling helps generate better adversarial examples.

We now turn to the two constraints imposed when constructing the candidate token set. Perhaps not surprisingly, ablating the textual similarity constraint (*w/o* sim) decreases the

Module	A-rate↑	Mod↓	PPL↓	GErr↓	Sim↑
TextFooler	56.1	23.3	331.3	1.43	0.69
CLARE	65.3	5.9	82.3	0.15	0.76
REPLACEONLY	58.8	7.9	85.6	0.11	0.75
INSERTONLY	59.4	6.9	94.8	0.20	0.76
MERGEONLY	21.0	6.2	95.2	0.01	0.79
w/o sim $> \ell$	70.0	5.4	80.9	0.11	0.72
w/o $p_{\text{MLM}} > k$	89.5	5.1	194.1	0.94	0.64

Table 5.4: Ablation study results. "*w/o* $p_{\text{MLM}} > k$" ablates the textual similarity constraint when constructing the candidate sets, while "*w/o* sim $> \ell$" ablates the masked language model probability constraint.

textual similarity performance, but increases others. Ablating the masked language model - *w/o* $p_{\text{MLM}} > k$ (Exhausting the vocabulary is computationally expensive. Therefore we randomly sample 200 tokens and then apply the similarity constraint to construct candidate set.), yields a better success rate, but much worse perplexity, grammaticality, and textual similarity.

Finally, we compare CLARE implemented with different masked language models. Table 5.5 summarizes the results. Overall, distilled RoBERTa performs the best, and BERT underperforms the others. Since the victim model is based on BERT, we conjecture that it is less efficient to attack a model using its own information.

5.5.2 Adversarial Training.

This section explores CLARE's potential in improving downstream models' accuracy and robustness. Following the adversarial training setup [132], we use CLARE to generate adversarial examples for AG news training instances, and include them as additional training

MLM	A-rate↑	Mod↓	PPL↓	GErr↓	Sim↑
$\text{RoBERTa}_{\text{distill}}$	65.3	5.9	82.3	0.15	0.76
$\text{RoBERTa}_{\text{base}}$	64.9	5.8	81.3	0.11	0.76
$\text{BERT}_{\text{base}}$	63.9	6.4	95.7	0.96	0.74

Table 5.5: Results of CLARE implemented with different masked language models (MLM).

Victim Model	Acc	A-rate	Mod
BERT (100% data)	95.0	65.3	5.9
+ 100% adversarial	-0.2	-23.4	+2.7
TextCNN (100% data)	91.2	93.8	6.5
+ 100% adversarial	-0.4	-10.2	+0.7
BERT (10% data)	92.5	84.0	5.4
+ 10% adversarial	-0.2	-14.4	+1.6
TextCNN (10% data)	83.6	97.3	6.2
+ 10% adversarial	+1.4	-3.7	+0.3

Table 5.6: Adversarial training results on AG News test set. "Acc" indicates accuracy.

data. We consider two settings: training with (1) full training data and full adversarial data and (2) 10% randomly-sampled training data and its adversarial data, to simulate the low-resource scenario. For both settings, we compare a BERT-based MLP classifier and a TextCNN ([63]) classifier without any pretrained embedding.

We first examine whether adversarial examples, as data augmentation, can help achieve better test accuracy. As shown in Table 5.6, when the full training data is available, adversarial training slightly *decreases* the test accuracy by 0.2% and 0.4% respectively. This

Replace	***Insert***	***Merge***
NOUN: 64%	*(NOUN, NOUN): 12%*	*ADJ-NOUN: 31%*
ADJ: 17%	*(ADJ, NOUN): 10%*	*NOUN-NOUN: 22%*
VERB: 7%	*(NOUN, VERB): 9%*	*DT-NOUN: 12%*

Context: ... Amit Yoran, the government's *cybersecurity* chief, abruptly resigned yesterday after a year ...

Replace: cybersecurity ← {*security, surveillance, cryptography, intelligence, encryption ...*}

Insert: cybersecurity __ chief ← {*technology, defense, intelligence, program, project ...*}

Merge: cybersecurity chief ← {*chief, consultant, administrator, scientist, secretary ...*}

Table 5.7: **Top**: Top-3 POS tags (or POS tag bigrams) and their percentages for each perturbation type. (a, b): insert a token between a and b. a-b: merge a and b into a token. **Bottom**: An AG news sample, where CLARE perturbs token "*cybersecurity*." Neither PWWS nor TextFooler is able to attack this token since it is out of their vocabularies.

aligns with previous observations [57]. When less training data is available, the BERT-based classifier has a similar accuracy drop. Interestingly, under the low-data scenario, TextCNN with adversarial training achieves better accuracy, with a 1.4% absolute improvement. This suggests that a model with less capacity can benefit more from silver data.

Does adversarial training help the models defend against adversarial attacks? In preliminary experiments, we found that it is more difficult to use other models to attack a victim model trained with the adversarial examples generated by CLARE, than to use CLARE itself. Thus, to evaluate this, we use CLARE to attack the classifiers trained with and without adversarial examples generated by itself.

A higher success rate and fewer modifications indicate a victim classifier is more vulnerable to adversarial attacks. As shown in Table 5.6, in 3 out of the 4 cases, adversarial training helps to decrease the attack success rate by more than 10.2%, and to increase the

number of modifications needed by more than 0.7. The only exception is the TextCNN model trained with 10% data. A possible reason could be that it is trained with few data and thus generalizes less well.

These results suggest that CLARE can be used to improve downstream models' robustness, with a negligible accuracy drop.

5.5.3 Perturbations by Part-of-speech Tags

In this section, we break down the adversarial attacks by part-of-speech (POS) tags. We find that most of the adversarial attacks happen to nouns or noun phrases. As shown in Table 5.7, 64% of the *Replace* actions are applied to nouns. *Insert* actions tend to insert tokens into noun phrase bigram: two of the most frequent POS bigrams are noun phrases. In fact, around 48% of the *Insert* actions are applied to noun phrases. This also justifies our choice of only applying *Merge* to noun phrases.

5.6 Conclusion

We have presented CLARE, a contextualized adversarial example generation model for text. It uses contextualized knowledge from pretrained masked language models, and can generate adversarial examples that are natural, fluent and grammatical. With three contextualized perturbation patterns, *Replace*, *Insert* and *Merge* in our arsenal, CLARE can produce outputs of varied lengths and achieves a higher attack success rate than baselines and with fewer edits. Human evaluation shows significant advantages of CLARE in terms of textual similarity, fluency and grammaticality.

AG (Sci&Tech)	Sprint Corp. is in talks with Qualcomm Inc. about using a network the chipmaker is building to deliver live television to Sprint mobile phone customers.
TextFooler (Business)	Sprint *Corps*. is in talks with Qualcomm Inc. about *operated* a network the chipmaker is *consolidation* to *doing viva* television to Sprint mobile phone customers.
CLARE (Business)	Sprint Corp. is in talks with Qualcomm Inc. about using a network Qualcomm is building to deliver *cable* television to Sprint mobile phone customers.
Yelp (Positive)	The food at this chain has always been consistently good. Our server in downtown (where we spent New Year's) was new, but that did not impact our service at all. She was prompt and attentive to our needs.
TextFooler (Negative)	The food at this chain has always been *necessarily ok*. Our server in downtown (where we spent New Year's) was new, but that did not impact our service at all. She was *early* and attentive to our needs.
CLARE (Negative)	The food at this chain has always been **looking** consistently good. Our server in downtown (where we spent New Year's) was new, but that did not *enhance* our service at all. She was prompt and attentive to our needs.
MNLI (Neutral)	*Premise*: Let me try it. She began snapping her fingers and saying the word eagerly, but nothing happened. *Hypothesis*: She became frustrated when the spell didn't work.
TextFooler (Contradiction)	*Premise*: *Authorisation* me *attempting* it. She *triggered flapping* her *pinkies* and *said* the word eagerly, but nothing *arisen*. *Hypothesis*: She became frustrated when the spell didn't work.
CLARE (Contradiction))	*Premise*: Let me try it. She began snapping her fingers and saying the word eagerly, but nothing **unexpected** happened. *Hypothesis*: She became frustrated when the spell didn't work.

Table 5.8: Adversarial examples produced by different models. The gold label of the original is shown below the (bolded) dataset name. ***Replace***, **Insert** and **Merge** are highlighted in *italic red*, **bold blue** and sans serif orange, respectively. (Best viewed in color).

QNLI (Entailment)	*Premise*: Who overturned the Taft Vale judgement ? *Hypothesis*: One of the first acts of the new Liberal Government was to reverse the Taff Vale judgement.
TextFooler (No Entailment)	*Premise*: Who overturned the Taft Vale judgement ? *Hypothesis*: One of the first acts of the new Liberal Government was to *invest* the Taff Vale judgement.
CLARE (No Entailment)	*Premise*: Who overturned the Taft Vale judgement ? *Hypothesis*: One of the first acts of the new Liberal *Constitution* was to reverse the Taff Vale judgement.
DBpedia (Transpor-tation)	Honda Crossroad. The Honda Crossroad refers to two specific types of SUVs made by Honda. One of them is a rebadged Land Rover Discovery Series I SUV while the other is a completely different vehicle introduced in 2008.
TextFooler (Album)	*Suzuki Junctions*. The *Suzuki* Crossroad refers to *three accurate typing* of *prius posed* by*Isuzu*. One of them is a rebadged Land Rover *Identify* Series I *LEXUS* while the other is a completely different vehicle introduced in 2008.
CLARE (Company)	Honda Crossroad. The Honda Crossroad refers to two specific *manufacturers* of SUVs made by Honda. One of them is a rebadged Land Rover Discovery Series I SUV while the other is a completely different vehicle introduced in 2008.
MRPC (*Para*)	*Premise*: The Securities Commission filed a civil fraud suit against the teen in Boston. *Hypothesis*: The Securities Commission brought a related civil case on Thursday.
TextFooler (No *para*)	*Premise*: The Securities Commission filed a civil fraud suit against the teen in Boston. *Hypothesis*: The Securities Commission brought a *connect* civil case on *Yesterday*.
CLARE (No *para*)	*Premise*: The Securities Commission filed a civil fraud suit against the teen in Boston. *Hypothesis*: The Securities Commission brought a *Massachusetts* civil *lawsuit* on Thursday.

Table 5.9: More generated adversarial examples. *Para* denotes *Paraphrase*.

Chapter 6

CONCLUSION

6.1 Summary

Due to the large variety and high complexity of natural language expressions, deep generative models still face challenges on natural language generation tasks. This thesis shows how we develop novel deep generative adversarial networks for generating high-quality text descriptions on different language generation tasks. Specifically, the contributions of this report are summarized as follows:

- In Chapter 2, we present a ranking-based generative adversarial network for generating high-quality natural language descriptions. The proposed deep generative model estimates the rewards through a relative ranking, relaxing the binary-classification restriction and conceiving a relative space with rich information for the discriminator in the adversarial learning framework. The relative ranking relieves the high-variance gradient problem in a high-dimensional generation space. The proposed generative model is generic and effective to synthesize various natural language sentences.

- In Chapter 3, we present a comparative adversarial learning network for generating diverse captions across images. The proposed network assesses the caption quality relatively by comparing other captions in an image-caption space. By suppressing the scores of image-mismatched captions in comparisons, the network can learn semantic relevance and generate captions with distinctiveness. We also propose a new caption diversity metric in the semantic level to evaluate the caption diversity.

- In Chapter 4, we show that most of existing works on text style transfer with limited

data may yield poor performance, where the generated text tends to use the most discriminative words that the target style prefers while ignoring the content. We explore two simple yet effective domain adaptive text style transfer models that leverage massively available data from other domains to achieve better content preservation and style control.

- In Chapter 5, we present a contextualized adversarial example generation model (CLARE) for texts. Previous rule-based adversarial generation models, such as replacing tokens with their synonyms, often have limitations to produce natural, fluent and grammatical outputs. CLARE modifies the input with three contextualized perturbations in a context-aware manner. The generated textual adversarial examples have a higher attack success rate and better preservation on textual similarity, fluency and grammaticality.

6.2 Future Works

Deep generative model is an emerging area to study in artificial intelligence. A few promising directions in natural language generation are worthy to explore in the future:

- **Connection between text style transfer and adversarial text generation.** Both text style transfer generation and textual adversarial example generation require suitable and minimal modifications on the original inputs while sustain conditional content preservation. Thus, it is interesting to explore the connection between the two tasks, e.g., whether we could generate adversarial examples on the text style transfer task to attack the style transfer model. On the other hand, the adversarial examples could help the text style transfer model to generate better style transferred sentences.

- **Controllable text generation.** Another direction worthy to explore is learning controllable latent variables in deep generative models. Deep Generative models typically learn latent variables from observed data, and generate samples drawing from the latent

variables. Learning meaningful and structural latent representations is a key step to generate sentences with expected attributes. In this manner, the generated sentences can be manipulated by the latent variables. This is an interesting and promising research area, where designing structural generative models or learning disentangled latent representations are valuable directions to be explored.

BIBLIOGRAPHY

[1] Reina Akama, Kazuaki Inada, Naoya Inoue, Sosuke Kobayashi, and Kentaro Inui. Generating stylistically consistent dialog responses with transfer learning. In *IJCNLP*, 2017.

[2] Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. Generating natural language adversarial examples. In *Proc. of EMNLP*, 2018.

[3] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. Spice: Semantic propositional image caption evaluation. In *Proc. ECCV*, pages 382–398, 2016.

[4] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*, 2018.

[5] Amittai Axelrod, Xiaodong He, and Jianfeng Gao. Domain adaptation via pseudo in-domain data selection. In *EMNLP*, 2011.

[6] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[7] Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proc. ACL workshops*, volume 29, pages 65–72, 2005.

[8] Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. *Proc. CoNLL*, page 10, 2016.

[9] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

[10] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*, 2018.

[11] Moitreya Chatterjee and Alexander G Schwing. Diverse and coherent paragraph generation from images. *arXiv preprint arXiv:1809.00681*, 2, 2018.

[12] 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.

[13] Liqun Chen, Shuyang Dai, Chenyang Tao, Haichao Zhang, Zhe Gan, Dinghan Shen, Yizhe Zhang, Guoyin Wang, Ruiyi Zhang, and Lawrence Carin. Adversarial text generation via feature-mover's distance. In *NeurIPS*, 2018.

[14] Yong Cheng, Lu Jiang, and Wolfgang Macherey. Robust neural machine translation with doubly adversarial inputs. In *Proc. of ACL*, 2019.

[15] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *EMNLP*, 2014.

[16] Chenhui Chu, Raj Dabre, and Sadao Kurohashi. An empirical comparison of simple domain adaptation methods for neural machine translation. *arXiv preprint arXiv:1701.03214*, 2017.

[17] Bo Dai and Dahua Lin. Contrastive learning for image captioning. In *NIPS*, pages 898–907, 2017.

[18] Bo Dai, Dahua Lin, Raquel Urtasun, and Sanja Fidler. Towards diverse and natural image descriptions via a conditional gan. *arXiv preprint arXiv:1703.06029*, 2017.

[19] Emily L Denton, Soumith Chintala, Rob Fergus, et al. Deep generative image models using a laplacian pyramid of adversarial networks. In *Proc. NIPS*, pages 1486–1494, 2015.

[20] Aditya Deshpande, Jyoti Aneja, Liwei Wang, Alexander Schwing, and David A Forsyth. Diverse and controllable image captioning with part-of-speech guidance. *arXiv preprint arXiv:1805.12589*, 2018.

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

[22] Qiming Diao, Minghui Qiu, Chao-Yuan Wu, Alexander J Smola, Jing Jiang, and Chong Wang. Jointly modeling aspects, ratings and sentiments for movie recommendation (jmars). In *SIGKDD*, 2014.

[23] William B Dolan and Chris Brockett. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing*, 2005.

[24] Cicero Nogueira dos Santos, Igor Melnyk, and Inkit Padhi. Fighting offensive language on social media with unsupervised text style transfer. In *ACL*, 2018.

[25] Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. Hotflip: White-box adversarial examples for text classification. In *Proc. of ACL*, 2018.

[26] Hao Fang, Saurabh Gupta, Forrest Iandola, Rupesh K Srivastava, Li Deng, Piotr Dollár, Jianfeng Gao, Xiaodong He, Margaret Mitchell, John C Platt, et al. From captions to visual concepts and back. In *CVPR*, 2015.

[27] William Fedus, Ian Goodfellow, and Andrew M Dai. Maskgan: better text generation via filling in the_. *arXiv preprint arXiv:1801.07736*, 2018.

[28] Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan. Style transfer in text: Exploration and evaluation. In *AAAI*, 2018.

[29] Chuang Gan, Zhe Gan, Xiaodong He, Jianfeng Gao, and Li Deng. Stylenet: Generating attractive visual captions with styles. In *CVPR*, 2017.

[30] Zhe Gan, Chuang Gan, Xiaodong He, Yunchen Pu, Kenneth Tran, Jianfeng Gao, Lawrence Carin, and Li Deng. Semantic compositional networks for visual captioning. In *CVPR*, 2017.

[31] Ji Gao, Jack Lanchantin, Mary Lou Soffa, and Yanjun Qi. Black-box generation of adversarial text sequences to evade deep learning classifiers. In *IEEE Security and Privacy Workshops (SPW)*, 2018.

[32] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1243–1252. JMLR. org, 2017.

[33] Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. Mask-predict: Parallel decoding of conditional masked language models. In *Proc. of EMNLP*, 2019.

[34] Hongyu Gong, Suma Bhat, Lingfei Wu, Jinjun Xiong, and Wen-mei Hwu. Reinforcement learning based text style transfer without parallel training corpus. *arXiv preprint arXiv:1903.10671*, 2019.

[35] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proc. NIPS*, pages 2672–2680, 2014.

[36] Ian J Goodfellow. On distinguishability criteria for estimating generative models. *arXiv preprint arXiv:1412.6515*, 2014.

[37] Ian J Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv preprint arXiv:1312.6211*, 2013.

[38] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *Proc. of ICLR*, 2015.

[39] Alex Graves. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, 2013.

[40] Arthur Gretton, AJ. Smola, Jiayuan Huang, Marcel Schmittfull, Karsten Borgwardt, and Bernhard Schölkopf. *Covariate shift and local learning by distribution matching.* MIT Press, 2009.

[41] Jiatao Gu, James Bradbury, Caiming Xiong, Victor OK Li, and Richard Socher. Non-autoregressive neural machine translation. In *Proc. of ICLR*, 2018.

[42] Jiaxian Guo, Sidi Lu, Han Cai, Weinan Zhang, Yong Yu, and Jun Wang. Long text generation via adversarial training with leaked information. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[43] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[44] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 1997.

[45] Zhiting Hu, Haoran Shi, Zichao Yang, Bowen Tan, Tiancheng Zhao, Junxian He, Wentao Wang, Lianhui Qin, Di Wang, et al. Texar: A modularized, versatile, and extensible toolkit for text generation. *arXiv preprint arXiv:1809.00794*, 2018.

[46] Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. Toward controlled generation of text. In *ICML*, 2017.

[47] Xinyu Hua and Lu Wang. A pilot study of domain adaptation effect for neural abstractive summarization. In *Proceedings of the Workshop on New Frontiers in Summarization*, 2017.

[48] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. Learning deep structured semantic models for web search using clickthrough data. In *Proc. CIKM*, pages 2333–2338, 2013.

[49] Ferenc Huszár. How (not) to train your generative model: Scheduled sampling, likelihood, adversary? *arXiv preprint arXiv:1511.05101*, 2015.

[50] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proc. CVPR*, 2017.

[51] Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. Adversarial example generation with syntactically controlled paraphrase networks. In *Proc. of NAACL*, 2018.

[52] Unnat Jain, Ziyu Zhang, and Alexander Schwing. Creativity: Generating diverse questions using variational autoencoders. In *CVPR*, 2017.

[53] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

[54] Mainak Jas and Devi Parikh. Image specificity. In *CVPR*, pages 2727–2736, 2015.

[55] Harsh Jhamtani, Varun Gangal, Eduard Hovy, and Eric Nyberg. Shakespearizing modern language using copy-enriched sequence to sequence models. In *Proceedings of the Workshop on Stylistic Variation*, 2017.

[56] Robin Jia and Percy Liang. Adversarial examples for evaluating reading comprehension systems. In *Proc. of EMNLP*, 2017.

[57] Robin Jia, Aditi Raghunathan, Kerem Göksel, and Percy Liang. Certified robustness to adversarial word substitutions. In *Proc. of EMNLP*, 2019.

[58] Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. Is bert really robust? natural language attack on text classification and entailment. In *Proc. of AAAI*, 2020.

[59] Thorsten Joachims. Optimizing search engines using clickthrough data. In *Proc. SIGKDD*, pages 133–142, 2002.

[60] Erik Jones, Robin Jia, Aditi Raghunathan, and Percy Liang. Robust encodings: A framework for combating adversarial typos. In *Proc. of ACL*, 2020.

[61] Katharina Kann, Sascha Rothe, and Katja Filippova. Sentence-level fluency evaluation: References help, but can be spared! In *Proc. of CNLP*, pages 313–323, 2018.

[62] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *CVPR*, pages 3128–3137, 2015.

[63] Yoon Kim. Convolutional neural networks for sentence classification. In *EMNLP*, 2014.

[64] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[65] Philipp Koehn and Josh Schroeder. Experiments in domain adaptation for statistical machine translation. In *Proceedings of the second workshop on statistical machine translation*, 2007.

[66] Keita Kurita, Paul Michel, and Graham Neubig. Weight poisoning attacks on pre-trained models. *arXiv preprint arXiv:2004.06660*, 2020.

[67] Matt J Kusner and José Miguel Hernández-Lobato. Gans for sequences of discrete elements with the gumbel-softmax distribution. *arXiv preprint arXiv:1611.04051*, 2016.

[68] Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. Unsupervised machine translation using monolingual corpora only. *arXiv preprint arXiv:1711.00043*, 2017.

[69] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. *arXiv preprint arXiv:1609.04802*, 2016.

[70] Jason Lee, Elman Mansimov, and Kyunghyun Cho. Deterministic non-autoregressive neural sequence modeling by iterative refinement. In *Proc. of EMNLP*, 2018.

[71] Jinfeng Li, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. Textbugger: Generating adversarial text against real-world applications. *arXiv preprint arXiv:1812.05271*, 2018.

[72] Jiwei Li, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. Visualizing and understanding neural models in nlp. In *Proc. of NAACL*, pages 681–691, 2016.

[73] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A diversity-promoting objective function for neural conversation models. *arXiv preprint arXiv:1510.03055*, 2015.

[74] Jiwei Li, Will Monroe, Tianlin Shi, Alan Ritter, and Dan Jurafsky. Adversarial learning for neural dialogue generation. *arXiv preprint arXiv:1701.06547*, 2017.

[75] Juncen Li, Robin Jia, He He, and Percy Liang. Delete, retrieve, generate: a simple approach to sentiment and style transfer. In *NAACL*, 2018.

[76] Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. Bert-attack: Adversarial attack against bert using bert. *arXiv preprint arXiv:2004.09984*, 2020.

[77] Bin Liang, Hongcheng Li, Miaoqiang Su, Pan Bian, Xirong Li, and Wenchang Shi. Deep text classification can be fooled. In *Proc. of IJCAI*, 2019.

[78] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *ACL Workshop*, 2004.

[79] Kevin Lin, Dianqi Li, Xiaodong He, Zhengyou Zhang, and Ming-Ting Sun. Adversarial ranking for language generation. In *Advances in Neural Information Processing Systems*, pages 3155–3165, 2017.

[80] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

[81] Siqi Liu, Zhenhai Zhu, Ning Ye, Sergio Guadarrama, and Kevin Murphy. Improved image captioning via policy gradient optimization of spider. In *Proceedings of the IEEE international conference on computer vision*, pages 873–881, 2017.

[82] Siqi Liu, Zhenhai Zhu, Ning Ye, Sergio Guadarrama, and Kevin Murphy. Improved image captioning via policy gradient optimization of spider. In *ICCV*, volume 3, 2017.

[83] Tie-Yan Liu et al. Learning to rank for information retrieval. *Foundations and Trends® in Information Retrieval*, 3(3):225–331, 2009.

[84] Xihui Liu, Hongsheng Li, Jing Shao, Dapeng Chen, and Xiaogang Wang. Show, tell and discriminate: Image captioning by self-retrieval with partially labeled data. *arXiv preprint arXiv:1803.08314*, 2018.

[85] 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 preprint arXiv:1907.11692*, 2019.

[86] Lajanugen Logeswaran, Honglak Lee, and Samy Bengio. Content preserving text generation with attribute controls. In *NeurIPS*, 2018.

[87] Ruotian Luo, Brian Price, Scott Cohen, and Gregory Shakhnarovich. Discriminability objective for training descriptive captions. In *CVPR*, pages 6964–6974, 2018.

[88] Xuezhe Ma, Chunting Zhou, Xian Li, Graham Neubig, and Eduard Hovy. Flowseq: Non-autoregressive conditional sequence generation with generative flow. In *Proc. of EMNLP*, 2019.

[89] Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016.

[90] Paul Michel and Graham Neubig. Extreme adaptation for personalized neural machine translation. In *ACL*, 2018.

[91] Ga Miller. Wordnet: A lexical database for english communications of the acm vol. 38. 1995.

[92] Remi Mir, Bjarke Felbo, Nick Obradovich, and Iyad Rahwan. Evaluating style transfer for text. *arXiv preprint arXiv:1904.02295*, 2019.

[93] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[94] John X Morris, Eli Lifland, Jack Lanchantin, Yangfeng Ji, and Yanjun Qi. Reevaluating adversarial examples in natural language. *arXiv preprint arXiv:2004.14174*, 2020.

[95] Nikola Mrkšić, Diarmuid Ó Séaghdha, Blaise Thomson, Milica Gasic, Lina M Rojas Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. Counter-fitting word vectors to linguistic constraints. In *Proc. of NAACL*, 2016.

[96] Daniel Naber et al. *A rule-based style and grammar checker*. Citeseer, 2003.

[97] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proc. ACL*, pages 311–318, 2002.

[98] Devi Parikh and Kristen Grauman. Relative attributes. In *Proc. ICCV*, pages 503–510, 2011.

[99] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Proc. of NeurIPS*, 2019.

[100] Hao Peng, Ankur P. Parikh, Manaal Faruqui, Bhuwan Dhingra, and Dipanjan Das. Text generation with exemplar-based adaptive decoding. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2019.

[101] Hao Peng, Roy Schwartz, Sam Thomson, and Noah A. Smith. Rational recurrences. In *In Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2018.

[102] Shrimai Prabhumoye, Yulia Tsvetkov, Ruslan Salakhutdinov, and Alan W Black. Style transfer through back-translation. In *ACL*, 2018.

[103] Danish Pruthi, Bhuwan Dhingra, and Zachary C Lipton. Combating adversarial misspellings with robust word recognition. In *Proc. of ACL*, 2019.

[104] Jipeng Qiang, Yun Li, Yi Zhu, and Yunhao Yuan. A simple bert-based approach for lexical simplification. In *Proc. of AAAI*, 2020.

[105] Xiaoye Qu, Zhikang Zou, Yu Cheng, Yang Yang, and Pan Zhou. Adversarial category alignment network for cross-domain sentiment classification. In *NAACL*, 2019.

[106] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[107] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9, 2019.

[108] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. In *Proc. of EMNLP*, 2016.

[109] Sudha Rao and Joel Tetreault. Dear sir or madam, may i introduce the gyafc dataset: Corpus, benchmarks and metrics for formality style transfer. In *NAACL*, 2018.

[110] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In *Proc. NIPS*, 2016.

[111] Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. Generating natural language adversarial examples through probability weighted word saliency. In *Proc. of ACL*, 2019.

[112] Yi Ren, Jinglin Liu, Xu Tan, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. A study of non-autoregressive model for sequence generation. *arXiv preprint arXiv:2004.10454*, 2020.

[113] Zhou Ren, Xiaoyu Wang, Ning Zhang, Xutao Lv, and Li-Jia Li. Deep reinforcement learning-based image captioning with embedding reward. In *CVPR*, 2017.

[114] Steven J. Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. In *CVPR*, 2017.

[115] Kevin Reschke, Adam Vogel, and Dan Jurafsky. Generating recommendation dialogs by extracting information from user reviews. In *ACL*, 2013.

[116] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Semantically equivalent adversarial rules for debugging nlp models. In *Proc. of ACL*, 2018.

[117] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *arXiv preprint arXiv:1606.03498*, 2016.

[118] Suranjana Samanta and Sameep Mehta. Towards crafting text adversarial samples. *arXiv preprint arXiv:1707.02812*, 2017.

[119] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. In *The 5th Workshop on Energy Efficient Machine Learning and Cognitive Computing, NeurIPS*, 2019.

[120] Stanislau Semeniuta, Aliaksei Severyn, and Sylvain Gelly. On accurate evaluation of gans for language generation. *arXiv preprint arXiv:1806.04936*, 2018.

[121] Rico Sennrich, Barry Haddow, and Alexandra Birch. Controlling politeness in neural machine translation via side constraints. In *NAACL*, 2016.

[122] Rico Sennrich, Barry Haddow, and Alexandra Birch. Improving neural machine translation models with monolingual data. In *ACL*, 2016.

[123] William Shakespeare. *The complete works of William Shakespeare*. Race Point Publishing, 2014.

[124] Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. Style transfer from non-parallel text by cross-alignment. In *NeurIPS*, 2017.

[125] Rakshith Shetty, Marcus Rohrbach, Lisa Anne Hendricks, Mario Fritz, and Bernt Schiele. Speaking the same language: Matching machine to human captions by adversarial training. In *ICCV*, 2017.

[126] Sandeep Subramanian, Guillaume Lample, Eric Michael Smith, Ludovic Denoyer, Marc'Aurelio Ranzato, and Y-Lan Boureau. Multiple-attribute text style transfer. *arXiv preprint arXiv:1811.00552*, 2018.

[127] Zhiqing Sun, Zhuohan Li, Haoqing Wang, Di He, Zi Lin, and Zhihong Deng. Fast structured decoding for sequence models. In *Proc. of NeurIPS*, 2019.

[128] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *NIPS*, 2014.

[129] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press Cambridge, 1998.

[130] Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *NIPS*, 2000.

[131] Richard S Sutton, David A McAllester, Satinder P Singh, Yishay Mansour, et al. Policy gradient methods for reinforcement learning with function approximation. In *NIPS*, volume 99, pages 1057–1063, 1999.

[132] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. In *Proc. of ICLR*, 2018.

[133] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.

[134] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *CVPR*, 2015.

[135] Ashwin K Vijayakumar, Michael Cogswell, Ramprasath R Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. Diverse beam search: Decoding diverse solutions from neural sequence models. *arXiv preprint arXiv:1610.02424*, 2016.

[136] Prashanth Vijayaraghavan and Deb Roy. Generating black-box adversarial examples for text classifiers using a deep reinforced model. *arXiv preprint arXiv:1909.07873*, 2019.

[137] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Proc. CVPR*, pages 3156–3164, 2015.

[138] Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. Universal adversarial triggers for attacking and analyzing nlp. In *Proc. of EMNLP*, 2019.

[139] 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 *Proc. of ICLR*, 2019.

[140] Boxin Wang, Hengzhi Pei, Han Liu, and Bo Li. Advcodec: Towards a unified framework for adversarial text generation. *arXiv preprint arXiv:1912.10375*, 2019.

[141] Liwei Wang, Alexander Schwing, and Svetlana Lazebnik. Diverse and accurate image description using a variational auto-encoder with an additive gaussian encoding space. In *NIPS*, pages 5756–5766, 2017.

[142] Xiaosen Wang, Hao Jin, and Kun He. Natural language adversarial attacks and defenses in word level. *arXiv preprint arXiv:1909.06723*, 2019.

[143] Zhuhao Wang, Fei Wu, Weiming Lu, Jun Xiao, Xi Li, Zitong Zhang, and Yueting Zhuang. Diverse image captioning via grouptalk. In *IJCAI*, 2016.

[144] Tsung-Hsien Wen, Milica Gašić, Nikola Mrkšić, Lina M Rojas-Barahona, Pei-Hao Su, David Vandyke, and Steve Young. Multi-domain neural network language generation for spoken dialogue systems. In *NAACL*, 2016.

[145] Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proc. of NAACL*, 2018.

[146] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

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

[148] Xing Wu, Shangwen Lv, Liangjun Zang, Jizhong Han, and Songlin Hu. Conditional bert contextual augmentation. In *Proc. of ICCS*, 2019.

[149] Xing Wu, Tao Zhang, Liangjun Zang, Jizhong Han, and Songlin Hu. "mask and infill": Applying masked language model to sentiment transfer. In *Proc. of IJCAI*, 2019.

[150] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

[151] Jingjing Xu, Sun Xu, Qi Zeng, Xiaodong Zhang, Xuancheng Ren, Houfeng Wang, and Wenjie Li. Unpaired sentiment-to-sentiment translation: A cycled reinforcement learning approach. In *ACL*, 2018.

[152] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *Proc. ICML*, pages 2048–2057, 2015.

[153] Zhen Yang, Wei Chen, Feng Wang, and Bo Xu. Improving neural machine translation with conditional sequence generative adversarial nets. *arXiv preprint arXiv:1703.04887*, 2017.

[154] Zichao Yang, Zhiting Hu, Chris Dyer, Eric P Xing, and Taylor Berg-Kirkpatrick. Unsupervised text style transfer using language models as discriminators. In *NeurIPS*, 2018.

[155] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. Seqgan: sequence generative adversarial nets with policy gradient. In *Proc. AAAI*, 2017.

[156] Yuan Zang, Fanchao Qi, Chenghao Yang, Zhiyuan Liu, Meng Zhang, Qun Liu, and Maosong Sun. Word-level textual adversarial attacking as combinatorial optimization. In *Proc. of ACL*, 2020.

[157] Huangzhao Zhang, Hao Zhou, Ning Miao, and Lei Li. Generating fluent adversarial examples for natural languages. In *Proc. of ACL*, 2019.

[158] Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. Personalizing dialogue agents: I have a dog, do you have pets too? In *ACL*, 2018.

[159] Wei Emma Zhang, Quan Z Sheng, Ahoud Alhazmi, and Chenliang Li. Adversarial attacks on deep-learning models in natural language processing: A survey. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(3):1–41, 2020.

[160] Xiang Zhang and Yann LeCun. Text understanding from scratch. *arXiv preprint arXiv:1502.01710*, 2015.

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

[162] Xingxing Zhang and Mirella Lapata. Chinese poetry generation with recurrent neural networks. In *Proc. EMNLP*, 2014.

[163] Ye Zhang, Nan Ding, and Radu Soricut. Shaped: Shared-private encoder-decoder for text style adaptation. In *NAACL*, 2018.

[164] Yi Zhang, Jingjing Xu, Pengcheng Yang, and Xu Sun. Learning sentiment memories for sentiment modification without parallel data. In *EMNLP*, 2018.

[165] Yizhe Zhang, Zhe Gan, Kai Fan, Zhi Chen, Ricardo Henao, Dinghan Shen, and Lawrence Carin. Adversarial feature matching for text generation. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 4006–4015. JMLR. org, 2017.

[166] Yizhe Zhang, Xiang Gao, Sungjin Lee, Chris Brockett, Michel Galley, Jianfeng Gao, and Bill Dolan. Consistent dialogue generation with self-supervised feature learning. *arXiv preprint arXiv:1903.05759*, 2019.

[167] Yizhe Zhang, Dinghan Shen, Guoyin Wang, Zhe Gan, Ricardo Henao, and Lawrence Carin. Deconvolutional paragraph representation learning. In *NeurIPS*, 2017.

[168] Yizhe Zhang, Guoyin Wang, Chunyuan Li, Zhe Gan, Chris Brockett, and Bill Dolan. Pointer: Constrained text generation via insertion-based generative pre-training. *arXiv preprint arXiv:2005.00558*, 2020.

[169] Zhengli Zhao, Dheeru Dua, and Sameer Singh. Generating natural adversarial examples. In *Proc. of ICLR*, 2018.

[170] Wangchunshu Zhou, Tao Ge, Ke Xu, Furu Wei, and Ming Zhou. Bert-based lexical substitution. In *Proc. of ACL*, 2019.

[171] Yichao Zhou, Jyun-Yu Jiang, Kai-Wei Chang, and Wei Wang. Learning to discriminate perturbations for blocking adversarial attacks in text classification. In *Proc. of EMNLP*, 2019.

[172] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.

[173] Wei Zou, Shujian Huang, Jun Xie, Xinyu Dai, and Jiajun Chen. A reinforced generation of adversarial samples for neural machine translation. In *Proc. of ACL*, 2020.