Language Style Transfer

by

Tianxiao Shen

B.Eng. in Computer Science and Technology
Tsinghua University, 2016

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2018

Author . . . .
Department of Electrical Engineering and Computer Science
May 21, 2018

Certified by . . . .
Regina Barzilay
Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by . . . .
Leslie A. Kolodziejski
Professor of Electrical Engineering and Computer Science
Chair, Department Committee on Graduate Students

Language Style Transfer

by

Tianxiao Shen

Submitted to the Department of Electrical Engineering and Computer Science
on May 21, 2018, in partial fulfillment of the
requirements for the degree of
Master of Science in Electrical Engineering and Computer Science

Abstract

This thesis studies style transfer on the basis of non-parallel text. This is an instance of a broad family of problems including machine translation, decipherment, and attribute modification. The key challenge is to separate the content from style in an unsupervised manner. We assume a shared latent content distribution across different text corpora, and propose a method that leverages refined alignment of latent representations to perform style transfer. The transferred sentences from one style should match example sentences from the other style as a population. To demonstrate the flexibility of the proposed model, we test it on three tasks: sentiment modification, decipherment of word substitution ciphers, and word order recovery. In both automatic and human evaluation our method achieves strong performance.

Thesis Supervisor: Regina Barzilay
Title: Professor of Electrical Engineering and Computer Science

Acknowledgments

I am extremely grateful to my advisor Regina Barzilay, whose vision, enthusiasm and research style have been playing a major role in my PhD journey and preparing me for becoming a researcher. Without her support, this work would not have started.

I am also very thankful to Tommi Jaakkola, who is always technically helpful and continuously brings me interesting and insightful inspirations. Working with him is a real pleasure.

It is a wonderful experience to work and study in the MIT NLP group, and I would like to thank all my current and former labmates. Special thanks to Tao Lei, my mentor and coauthor, whose patient and experienced guidance has helped me greatly in my initial exploration in NLP.

Finally, deepest thanks to my parents for their unlimited love and encouragement. This thesis is dedicated to them.

Bibliographic Note

Portions of this thesis are based on prior peer-reviewed publication Shen et al. [48].

The code and data of the work presented in this thesis are available at `https://github.com/shentianxiao/language-style-transfer`.

Contents

1 Introduction 15

1.1 Motivation 15

1.2 Contributions 16

1.3 Outline 17

2 Formulation 19

2.1 Example 1: Gaussian 20

2.2 Example 2: Word substitution 21

3 Method 23

3.1 Aligned auto-encoder 24

3.2 Cross-aligned auto-encoder 25

4 Experiments 29

4.1 Setup 29

4.2 Results 31

5 Related work 35

5.1 Style transfer in vision 35

5.2 Neural networks for text generation 36

5.3 Adversarial training over discrete samples 37

6 Conclusions 39

List of Figures

3-1 An overview of the proposed cross-alignment method. $\mathcal{X}_1$ and $\mathcal{X}_2$ are two sentence domains with different styles y_1 and y_2, and $\mathcal{Z}$ is the shared latent content space. Encoder E maps a sentence to its content representation, and generator G generates the sentence back when combining with the original style. When combining with a different style, transferred $\tilde{\mathcal{X}}_1$ is aligned with $\mathcal{X}_2$ and $\tilde{\mathcal{X}}_2$ is aligned with $\mathcal{X}_1$ at the distributional level. 26

3-2 Cross-aligning between $\boldsymbol{x}_1$ and transferred $\boldsymbol{x}_2$. For $\boldsymbol{x}_1$, G is teacher-forced by its words $\boldsymbol{w}_1\boldsymbol{w}_2\cdots\boldsymbol{w}_t$. For transfered $\boldsymbol{x}_2$, G is self-fed by previous output logits. The sequence of hidden states $\boldsymbol{h}^0, \cdots, \boldsymbol{h}^t$ and $\tilde{\boldsymbol{h}}^0, \cdots, \tilde{\boldsymbol{h}}^t$ are passed to discriminator D_1 to be aligned. Note that our first variant aligned auto-encoder is a special case of this, where only $\boldsymbol{h}^0$ and $\tilde{\boldsymbol{h}}^0$, i.e. $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$, are aligned. 27

List of Tables

4.1 Sentiment accuracy of transferred sentences, as measured by a pre-trained classifier. 32

4.2 Human evaluations on sentiment, fluency and overall transfer quality. Fluency rating is from 1 (unreadable) to 4 (perfect). Overall transfer quality is evaluated in a comparative manner, where the judge is shown a source sentence and two transferred sentences, and decides whether they are both good, both bad, or one is better. 32

4.3 Sentiment transfer samples. The first line is an input sentence, the second and third lines are the generated sentences after sentiment transfer by Hu et al. [23] and our cross-aligned auto-encoder, respectively. . . 33

4.4 Bleu scores of word substitution decipherment and word order recovery. 33

Chapter 1

Introduction

1.1 Motivation

Using massive amounts of parallel data has been essential for recent advances in text generation tasks, such as machine translation and summarization. However, in many text generation problems, we can only assume access to non-parallel or mono-lingual data. Problems such as decipherment or style transfer are all instances of this family of tasks. In all of these problems, we must preserve the content of the source sentence but render the sentence consistent with desired presentation constraints (e.g., style, plaintext/ciphertext).

The goal of controlling one aspect of a sentence such as style independently of its content requires that we can disentangle the two. However, these aspects interact in subtle ways in natural language sentences, and we can succeed in this task only approximately even in the case of parallel data. Our task is more challenging here. We merely assume access to two corpora of sentences with the same distribution of content albeit rendered in different styles. Our goal is to demonstrate that this distributional equivalence of content, if exploited carefully, suffices for us to learn to map a sentence in one style to a style-independent content vector and then decode it to a sentence with the same content but a different style.

We consider the most general setting: given two corpora of different styles $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$, learn to transfer from one style to the other. The style can be formal/informal, plain/poetic, serious/humorous, democratic/republican, different personal styles, etc.

It has a wide range of applications, such as to design personalized chatbots, and to appropriately convey a message according to different social contexts. Moreover, enabling machines to distinguish and manipulate the style and content of language is an important step towards real language understanding.

1.2 Contributions

The primary contributions of this thesis are threefold:

- We make the first step to study language style transfer from non-parallel text. We formulate this problem into a generative framework where style and content are latent variables. We investigate under what conditions this unsupervised learning is feasible, and provide theoretical justifications that enlighten our model design.

- We propose a neural model that learns from non-parallel data to perform language style transfer. We derive constraints that transferred sentences from one style should match example sentences from the other style as a population. We use adversarial training to ensure these distributional constraints, and introduce a refined alignment of sentence representations to facilitate the discrete comparison.

- We design various experiments to both quantitatively and qualitatively evaluate language style transfer systems. Our method is flexible and capable for different kinds of style transfer, demonstrated by its success on sentiment modification, decipherment of word substitution ciphers, and word order recovery.

We believe our work would promote style transfer research in natural language processing (NLP), as there are plenty of interesting and important theoretical and practical problems to explore. Our approach merely assumes access to non-parallel data, opening up possibilities for many text generation tasks where parallel corpora are very costly to collect or even do not exist.

1.3 Outline

The rest of this thesis is organized as follows:

- **Chapter 2** describes our formulation of non-parallel style transfer in a latent variable generative framework, and provides theoretical analysis that illustrates important properties of this problem.

- **Chapter 3** presents our models aligned auto-encoder and cross-aligned auto-encoder with the associated training algorithm.

- **Chapter 4** detail our experimental design to comprehensively evaluate our proposed method and baselines.

- **Chapter 5** summarizes related work in image style transfer, neural networks for text generation, adversarial training and back-propagation through discrete samples.

- **Chapter 6** concludes the thesis and discusses directions for future work.

Chapter 2

Formulation

In this chapter, we formalize the task of non-parallel style transfer and discuss the feasibility of the learning problem. We assume the data are generated by the following process:

1. a latent style variable $\boldsymbol{y}$ is generated from some distribution $p(\boldsymbol{y})$;

2. a latent content variable $\boldsymbol{z}$ is generated from some distribution $p(\boldsymbol{z})$;

3. a datapoint $\boldsymbol{x}$ is generated from conditional distribution $p(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{z})$.

We observe two datasets with the same content distribution but different styles $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$, where $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$ are unknown. Specifically, the two observed datasets $\boldsymbol{X}_1 = \{\boldsymbol{x}_1^{(1)}, \cdots, \boldsymbol{x}_1^{(n)}\}$ and $\boldsymbol{X}_2 = \{\boldsymbol{x}_2^{(1)}, \cdots, \boldsymbol{x}_2^{(m)}\}$ consist of samples drawn from $p(\boldsymbol{x}_1|\boldsymbol{y}_1)$ and $p(\boldsymbol{x}_2|\boldsymbol{y}_2)$ respectively. We want to estimate the style transfer functions between them, namely $p(\boldsymbol{x}_1|\boldsymbol{x}_2; \boldsymbol{y}_1, \boldsymbol{y}_2)$ and $p(\boldsymbol{x}_2|\boldsymbol{x}_1; \boldsymbol{y}_1, \boldsymbol{y}_2)$.

A question we must address is when this estimation problem is feasible. Essentially, we only observe the marginal distributions of $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, yet we are going to recover their joint distribution:

$$p(\boldsymbol{x}_1, \boldsymbol{x}_2|\boldsymbol{y}_1, \boldsymbol{y}_2) = \int_{\boldsymbol{z}} p(\boldsymbol{z})p(\boldsymbol{x}_1|\boldsymbol{y}_1, \boldsymbol{z})p(\boldsymbol{x}_2|\boldsymbol{y}_2, \boldsymbol{z})d\boldsymbol{z} \tag{2.1}$$

As we only observe $p(\boldsymbol{x}_1|\boldsymbol{y}_1)$ and $p(\boldsymbol{x}_2|\boldsymbol{y}_2)$, $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$ are unknown to us. If two different $\boldsymbol{y}$ and $\boldsymbol{y}'$ lead to the same distribution $p(\boldsymbol{x}|\boldsymbol{y}) = p(\boldsymbol{x}|\boldsymbol{y}')$, then given a dataset $\boldsymbol{X}$ sampled from it, its underlying style can be either $\boldsymbol{y}$ or $\boldsymbol{y}'$. Consider the following two cases: (1) both datasets $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ are sampled from the same style $\boldsymbol{y}$; (2) $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ are sampled from style $\boldsymbol{y}$ and $\boldsymbol{y}'$ respectively. These two scenarios have different joint distributions, but the observed marginal distributions are the same. To prevent such confusion, we constrain the underlying distributions as stated in the following proposition:

Proposition 1. *In the generative framework above, $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$'s joint distribution can be recovered from their marginals only if for any different $\boldsymbol{y}, \boldsymbol{y}' \in \mathcal{Y}$, distributions $p(\boldsymbol{x}|\boldsymbol{y})$ and $p(\boldsymbol{x}|\boldsymbol{y}')$ are different.*

This proposition basically says that $\boldsymbol{X}$ generated from different styles should be "distinct" enough, otherwise the transfer task between styles is not well defined. While this seems trivial, it may not hold even for simplified data distributions. The following examples illustrate how the transfer (and recovery) becomes feasible or infeasible under different model assumptions. As we shall see, for a certain family of styles $\mathcal{Y}$, the more complex distribution for $\boldsymbol{z}$, the more probable it is to recover the transfer function and the easier it is to search for the transfer.

2.1 Example 1: Gaussian

Consider the common choice that $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ has a centered isotropic Gaussian distribution. Suppose a style $\boldsymbol{y} = (\boldsymbol{A}, \boldsymbol{b})$ is an affine transformation, i.e. $\boldsymbol{x} = \boldsymbol{A}\boldsymbol{z} + \boldsymbol{b} + \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon}$ is a noise variable. For $\boldsymbol{b} = \boldsymbol{0}$ and any orthogonal matrix $\boldsymbol{A}$, $\boldsymbol{A}\boldsymbol{z} + \boldsymbol{b} \sim N(\boldsymbol{0}, \boldsymbol{I})$ and hence $\boldsymbol{x}$ has the same distribution for any such styles $\boldsymbol{y} = (\boldsymbol{A}, \boldsymbol{0})$. In this case, the effect of rotation cannot be recovered.

Interestingly, if $\boldsymbol{z}$ has **a more complex distribution**, such as a Gaussian mixture, then affine transformations can be uniquely determined.

Lemma 1. *Let $\boldsymbol{z}$ be a mixture of Gaussians $p(\boldsymbol{z}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{z}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$. Assume $K \geq 2$, and there are two different $\boldsymbol{\Sigma}_i \neq \boldsymbol{\Sigma}_j$. Let $\mathcal{Y} = \{(\boldsymbol{A}, \boldsymbol{b}) || \boldsymbol{A}| \neq 0\}$ be all invertible affine transformations, and $p(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{z}) = \mathcal{N}(\boldsymbol{x}; \boldsymbol{A}\boldsymbol{z} + \boldsymbol{b}, \epsilon^2 \boldsymbol{I})$, in which ϵ is a noise. Then for all $\boldsymbol{y} \neq \boldsymbol{y}' \in \mathcal{Y}$, $p(\boldsymbol{x}|\boldsymbol{y})$ and $p(\boldsymbol{x}|\boldsymbol{y}')$ are different distributions.*

Proof.

$$p(\boldsymbol{x}|\boldsymbol{y} = (\boldsymbol{A}, \boldsymbol{b})) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}; \boldsymbol{A}\boldsymbol{\mu}_k + \boldsymbol{b}, \boldsymbol{A}\boldsymbol{\Sigma}_k \boldsymbol{A}^\top + \epsilon^2 \boldsymbol{I})$$

For different $\boldsymbol{y} = (\boldsymbol{A}, \boldsymbol{b})$ and $\boldsymbol{y'} = (\boldsymbol{A'}, \boldsymbol{b'})$, $p(\boldsymbol{x}|\boldsymbol{y}) = p(\boldsymbol{x}|\boldsymbol{y'})$ entails that for $k = 1, \cdots, K$,

$$\begin{cases} \boldsymbol{A}\boldsymbol{\mu}_k + \boldsymbol{b} = \boldsymbol{A'}\boldsymbol{\mu}_k + \boldsymbol{b'} \\ \boldsymbol{A}\boldsymbol{\Sigma}_k\boldsymbol{A}^\top = \boldsymbol{A'}\boldsymbol{\Sigma}_k\boldsymbol{A'}^\top \end{cases}$$

Since all $\mathcal{Y}$ are invertible,

$$(\boldsymbol{A}^{-1}\boldsymbol{A'})\boldsymbol{\Sigma}_k(\boldsymbol{A}^{-1}\boldsymbol{A'})^\top = \boldsymbol{\Sigma}_k$$

Suppose $\boldsymbol{\Sigma}_k = \boldsymbol{Q}_k\boldsymbol{D}_k\boldsymbol{Q}_k^\top$ is $\boldsymbol{\Sigma}_k$'s orthogonal diagonalization. If $k = 1$, all solutions for $\boldsymbol{A}^{-1}\boldsymbol{A'}$ have the form:

$$\left\{\boldsymbol{Q}\boldsymbol{D}^{1/2}\boldsymbol{U}\boldsymbol{D}^{-1/2}\boldsymbol{Q}^\top \middle| \boldsymbol{U} \text{ is orthogonal}\right\}$$

However, when $K \geq 2$ and there are two different $\boldsymbol{\Sigma}_i \neq \boldsymbol{\Sigma}_j$, the only solution is $\boldsymbol{A}^{-1}\boldsymbol{A'} = \boldsymbol{I}$, i.e. $\boldsymbol{A} = \boldsymbol{A'}$, and thus $\boldsymbol{b} = \boldsymbol{b'}$.

Therefore, for all $\boldsymbol{y} \neq \boldsymbol{y'}$, $p(\boldsymbol{x}|\boldsymbol{y}) \neq p(\boldsymbol{x}|\boldsymbol{y'})$. □

Theorem 1. *If the distribution of $\boldsymbol{z}$ is a mixture of Gaussians which has more than two different components, and $\boldsymbol{x}_1, \boldsymbol{x}_2$ are two affine transformations of $\boldsymbol{z}$, then the transfer between them can be recovered given their respective marginals.*

2.2 Example 2: Word substitution

Consider here another example when $\boldsymbol{z}$ is a bi-gram language model and a style $\boldsymbol{y}$ is a vocabulary in use that maps each "content word" onto its surface form (lexical form). If we observe two realizations $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ of the same language $\boldsymbol{z}$, the transfer and recovery problem becomes inferring a word alignment between $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$.

Note that this is a simplified version of language decipherment or translation. Nevertheless, the recovery problem is still sufficiently hard. To see this, let $\boldsymbol{M}_1, \boldsymbol{M}_2 \in \mathcal{R}^{n\times n}$ be the estimated bi-gram probability matrix of data $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ respectively. Seeking the word alignment is equivalent to finding a permutation matrix $\boldsymbol{P}$ such that $\boldsymbol{P}^\top \boldsymbol{M}_1 \boldsymbol{P} \approx \boldsymbol{M}_2$, which can be expressed as an optimization problem,

$$\min_{\boldsymbol{P}} \ \|\boldsymbol{P}^\top \boldsymbol{M}_1 \boldsymbol{P} - \boldsymbol{M}_2\|^2$$

The same formulation applies to graph isomorphism (GI) problems given $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$ as the adjacency matrices of two graphs, suggesting that determining the existence and uniqueness of $\boldsymbol{P}$ is at least GI hard. Fortunately, if $\boldsymbol{M}$ as a graph is complex enough, the search problem could be more tractable. For instance, if each vertex's weights of incident edges as a set is unique, then finding the isomorphism can be done by simply matching the sets of edges. This assumption largely applies to our scenario where $\boldsymbol{z}$ is a complex language model. We empirically demonstrate this in the results chapter.

The above examples suggest that $\boldsymbol{z}$ as the latent content variable should carry most complexity of data $\boldsymbol{x}$, while $\boldsymbol{y}$ as the latent style variable should have relatively simple effects. We construct the model accordingly in the next chapter.

Chapter 3

Method

Learning the style transfer function under our generative assumption is essentially learning the conditional distribution $p(\boldsymbol{x}_1|\boldsymbol{x}_2;\boldsymbol{y}_1,\boldsymbol{y}_2)$ and $p(\boldsymbol{x}_2|\boldsymbol{x}_1;\boldsymbol{y}_1,\boldsymbol{y}_2)$. Unlike in vision where images are continuous and hence the transfer functions can be learned and optimized directly, the discreteness of language requires us to operate through the latent space. Since $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are conditionally independent given the latent content variable $\boldsymbol{z}$,

$$\begin{aligned} p(\boldsymbol{x}_1|\boldsymbol{x}_2;\boldsymbol{y}_1,\boldsymbol{y}_2) &= \int_{\boldsymbol{z}} p(\boldsymbol{x}_1,\boldsymbol{z}|\boldsymbol{x}_2;\boldsymbol{y}_1,\boldsymbol{y}_2)d\boldsymbol{z} \\ &= \int_{\boldsymbol{z}} p(\boldsymbol{z}|\boldsymbol{x}_2,\boldsymbol{y}_2)\cdot p(\boldsymbol{x}_1|\boldsymbol{y}_1,\boldsymbol{z})d\boldsymbol{z} \\ &= \mathbb{E}_{\boldsymbol{z}\sim p(\boldsymbol{z}|\boldsymbol{x}_2,\boldsymbol{y}_2)}[p(\boldsymbol{x}_1|\boldsymbol{y}_1,\boldsymbol{z})] \end{aligned} \tag{3.1}$$

This suggests us learning an auto-encoder model. Specifically, a style transfer from $\boldsymbol{x}_2$ to $\boldsymbol{x}_1$ involves two steps—an encoding step that infers $\boldsymbol{x}_2$'s content $\boldsymbol{z}\sim p(\boldsymbol{z}|\boldsymbol{x}_2,\boldsymbol{y}_2)$, and a decoding step which generates the transferred counterpart from $p(\boldsymbol{x}_1|\boldsymbol{y}_1,\boldsymbol{z})$. In this work, we approximate and train $p(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{y})$ and $p(\boldsymbol{x}|\boldsymbol{y},\boldsymbol{z})$ using neural networks (where $\boldsymbol{y}\in\{\boldsymbol{y}_1,\boldsymbol{y}_2\}$).

Let $E:\mathcal{X}\times\mathcal{Y}\to\mathcal{Z}$ be an encoder that infers the content $\boldsymbol{z}$ for a given sentence $\boldsymbol{x}$ and a style $\boldsymbol{y}$, and $G:\mathcal{Y}\times\mathcal{Z}\to\mathcal{X}$ be a generator that generates a sentence $\boldsymbol{x}$ from a given style $\boldsymbol{y}$ and content $\boldsymbol{z}$. E and G form an auto-encoder when applying to the same style, and thus we have reconstruction loss,

$$\begin{aligned}\mathcal{L}_{\text{rec}}(\boldsymbol{\theta}_E, \boldsymbol{\theta}_G) = \; & \mathbb{E}_{\boldsymbol{x}_1 \sim \boldsymbol{X}_1}[-\log p_G(\boldsymbol{x}_1 | \boldsymbol{y}_1, E(\boldsymbol{x}_1, \boldsymbol{y}_1))] + \\ & \mathbb{E}_{\boldsymbol{x}_2 \sim \boldsymbol{X}_2}[-\log p_G(\boldsymbol{x}_2 | \boldsymbol{y}_2, E(\boldsymbol{x}_2, \boldsymbol{y}_2))]\end{aligned} \tag{3.2}$$

where $\boldsymbol{\theta}$ are the parameters to estimate.

In order to make a meaningful transfer by flipping the style, $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$'s content space must coincide, as our generative framework presumed. To constrain that $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are generated from the same latent content distribution $p(\boldsymbol{z})$, one option is to apply a variational auto-encoder [28]. A VAE imposes a prior density $p(\boldsymbol{z})$, such as $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$, and uses a KL-divergence regularizer to align both posteriors $p_E(\boldsymbol{z}|\boldsymbol{x}_1, \boldsymbol{y}_1)$ and $p_E(\boldsymbol{z}|\boldsymbol{x}_2, \boldsymbol{y}_2)$ to it,

$$\mathcal{L}_{\text{KL}}(\boldsymbol{\theta}_E) = \; \mathbb{E}_{\boldsymbol{x}_1 \sim \boldsymbol{X}_1}[D_{\text{KL}}(p_E(\boldsymbol{z}|\boldsymbol{x}_1, \boldsymbol{y}_1) \| p(\boldsymbol{z}))] + \mathbb{E}_{\boldsymbol{x}_2 \sim \boldsymbol{X}_2}[D_{\text{KL}}(p_E(\boldsymbol{z}|\boldsymbol{x}_2, \boldsymbol{y}_2) \| p(\boldsymbol{z}))] \tag{3.3}$$

The overall objective is to minimize $\mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{KL}}$, which is the evidence lower bound (ELBO) of data likelihood.

However, as we have argued in the previous chapter, restricting $\boldsymbol{z}$ to a simple and even distribution and pushing most complexity to the decoder may not be a good strategy for non-parallel style transfer. In contrast, a standard auto-encoder simply minimizes the reconstruction error, encouraging $\boldsymbol{z}$ to carry as much information about $\boldsymbol{x}$ as possible. On the other hand, it lowers the entropy in $p(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{z})$, which helps to produce meaningful style transfer in practice as we flip between $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$. Without explicitly modeling $p(\boldsymbol{z})$, it is still possible to force distributional alignment of $p(\boldsymbol{z}|\boldsymbol{y}_1)$ and $p(\boldsymbol{z}|\boldsymbol{y}_2)$. To this end, we introduce two constrained variants of auto-encoder.

3.1 Aligned auto-encoder

Dispense with VAEs that make an explicit assumption about $p(\boldsymbol{z})$ and align both posteriors to it, we align $p_E(\boldsymbol{z}|\boldsymbol{y}_1)$ and $p_E(\boldsymbol{z}|\boldsymbol{y}_2)$ with each other, which leads to the following constrained optimization problem:

$$\begin{aligned} \boldsymbol{\theta}^* &= \arg\min_{\boldsymbol{\theta}} \mathcal{L}_{\text{rec}}(\boldsymbol{\theta}_E, \boldsymbol{\theta}_G) \\ \text{s.t.} \quad E(\boldsymbol{x}_1, \boldsymbol{y}_1) &\overset{\text{d}}{=} E(\boldsymbol{x}_2, \boldsymbol{y}_2) \qquad \boldsymbol{x}_1 \sim \boldsymbol{X}_1, \boldsymbol{x}_2 \sim \boldsymbol{X}_2 \end{aligned} \tag{3.4}$$

In practice, a Lagrangian relaxation of the primal problem is instead optimized. We introduce an adversarial discriminator D to align the aggregated posterior distribution of $\boldsymbol{z}$ from different styles [39]. D aims to distinguish between these two distributions:

$$\mathcal{L}_{\text{adv}}(\boldsymbol{\theta}_E, \boldsymbol{\theta}_D) = \mathbb{E}_{\boldsymbol{x}_1 \sim \boldsymbol{X}_1}[-\log D(E(\boldsymbol{x}_1, \boldsymbol{y}_1))] + \mathbb{E}_{\boldsymbol{x}_2 \sim \boldsymbol{X}_2}[-\log(1 - D(E(\boldsymbol{x}_2, \boldsymbol{y}_2)))] \tag{3.5}$$

The overall training objective is a min-max game played among the encoder E, generator G and discriminator D. They constitute an aligned auto-encoder:

$$\min_{E,G} \max_{D} \mathcal{L}_{\text{rec}} - \lambda \mathcal{L}_{\text{adv}} \tag{3.6}$$

We implement the encoder E and generator G using single-layer RNNs with GRU cell. E takes an input sentence $\boldsymbol{x}$ with initial hidden state $\boldsymbol{y}$, and outputs the last hidden state $\boldsymbol{z}$ as its content representation. G generates a sentence $\boldsymbol{x}$ conditioned on latent state $(\boldsymbol{y}, \boldsymbol{z})$. To align the distributions of $\boldsymbol{z}_1 = E(\boldsymbol{x}_1, \boldsymbol{y}_1)$ and $\boldsymbol{z}_2 = E(\boldsymbol{x}_2, \boldsymbol{y}_2)$, the discriminator D is a feed-forward network with a single hidden layer and a sigmoid output layer.

3.2 Cross-aligned auto-encoder

The second variant, cross-aligned auto-encoder, directly aligns the transfered samples from one style with the true samples from the other. Under the generative assumption,

$$p(\boldsymbol{x}_2|\boldsymbol{y}_2) = \int_{\boldsymbol{x}_1} p(\boldsymbol{x}_2|\boldsymbol{x}_1; \boldsymbol{y}_1, \boldsymbol{y}_2) p(\boldsymbol{x}_1|\boldsymbol{y}_1) d\boldsymbol{x}_1 \tag{3.7}$$

thus $\boldsymbol{x}_2$ (sampled from the left-hand side) should exhibit the same distribution as transferred $\boldsymbol{x}_1$ (sampled from the right-hand side), and vice versa. Similar to our first

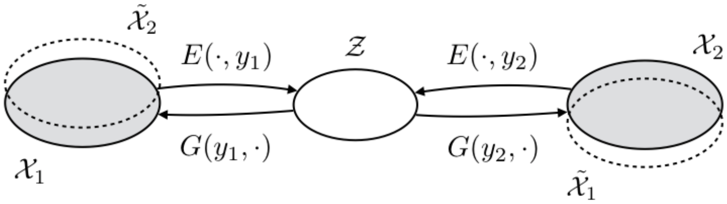

Figure 3-1: An overview of the proposed cross-alignment method. $\mathcal{X}_1$ and $\mathcal{X}_2$ are two sentence domains with different styles y_1 and y_2, and $\mathcal{Z}$ is the shared latent content space. Encoder E maps a sentence to its content representation, and generator G generates the sentence back when combining with the original style. When combining with a different style, transferred $\tilde{\mathcal{X}_1}$ is aligned with $\mathcal{X}_2$ and $\tilde{\mathcal{X}_2}$ is aligned with $\mathcal{X}_1$ at the distributional level.

model, the second model uses two discriminators D_1 and D_2 to align the populations. D_1's job is to distinguish between real $\boldsymbol{x}_1$ and transferred $\boldsymbol{x}_2$, and D_2's job is to distinguish between real $\boldsymbol{x}_2$ and transferred $\boldsymbol{x}_1$.

Adversarial training over the discrete samples generated by G hinders gradients propagation. Although sampling-based gradient estimator such as REINFORCE [52] can by adopted, training with these methods can be unstable due to the high variance of the sampled gradient. Instead, we employ two recent techniques to approximate the discrete training [23, 33]. First, instead of feeding a single sampled word as the input to the generator RNN, we use the softmax distribution over words instead. Specifically, during the generating process of transferred $\boldsymbol{x}_2$ from $G(\boldsymbol{y}_1, \boldsymbol{z}_2)$, suppose at time step t the output logit vector is $\boldsymbol{v}_t$. We feed its peaked distribution $\text{softmax}(\boldsymbol{v}_t/\gamma)$ as the next input, where $\gamma \in (0, 1)$ is a temperature parameter.

Secondly, we use Professor-Forcing [33] to match the sequence of hidden states instead of the output words, which contains the information about outputs and is smoothly distributed. That is, the input to the discriminator D_1 is the sequence of hidden states of either (1) $G(\boldsymbol{y}_1, \boldsymbol{z}_1)$ teacher-forced by a real example $\boldsymbol{x}_1$, or (2) $G(\boldsymbol{y}_1, \boldsymbol{z}_2)$ self-fed by previous soft distributions.

The running procedure of our cross-aligned auto-encoder is illustrated in Figure 3-2. Note that cross-aligning strengthens the alignment of latent variable $\boldsymbol{z}$ over the recurrent network of generator G. By aligning the whole sequence of hidden states, it prevents $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$'s initial misalignment from propagating through the recurrent generating process, as a result of which the transferred sentence may end up somewhere

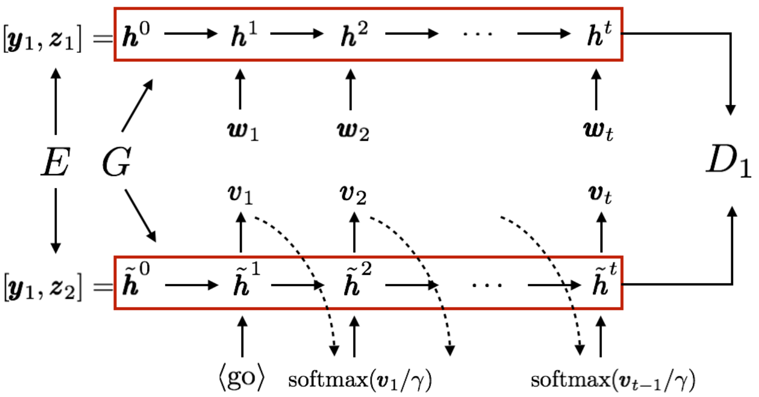
$[\boldsymbol{y}_1, \boldsymbol{z}_1] = \boldsymbol{h}^0$
$\boldsymbol{h}^1$
$\boldsymbol{h}^2$
$\cdots$
$\boldsymbol{h}^t$
$\boldsymbol{w}_1$
$\boldsymbol{w}_2$
$\boldsymbol{w}_t$
E
G
D_1
$\boldsymbol{v}_1$
$\boldsymbol{v}_2$
$\boldsymbol{v}_t$
$[\boldsymbol{y}_1, \boldsymbol{z}_2] = \tilde{\boldsymbol{h}}^0$
$\tilde{\boldsymbol{h}}^1$
$\tilde{\boldsymbol{h}}^2$
$\cdots$
$\tilde{\boldsymbol{h}}^t$
$\langle \mathrm{go} \rangle$
$\mathrm{softmax}(\boldsymbol{v}_1/\gamma)$
$\mathrm{softmax}(\boldsymbol{v}_{t-1}/\gamma)$

Figure 3-2: Cross-aligning between $\boldsymbol{x}_1$ and transferred $\boldsymbol{x}_2$. For $\boldsymbol{x}_1$, G is teacher-forced by its words $\boldsymbol{w}_1\boldsymbol{w}_2\cdots\boldsymbol{w}_t$. For transfered $\boldsymbol{x}_2$, G is self-fed by previous output logits. The sequence of hidden states $\boldsymbol{h}^0, \cdots, \boldsymbol{h}^t$ and $\tilde{\boldsymbol{h}}^0, \cdots, \tilde{\boldsymbol{h}}^t$ are passed to discriminator D_1 to be aligned. Note that our first variant aligned auto-encoder is a special case of this, where only $\boldsymbol{h}^0$ and $\tilde{\boldsymbol{h}}^0$, i.e. $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$, are aligned.

far from the target domain.

We implement both D_1 and D_2 using convolutional neural networks for sequence classification [27]. The training algorithm is presented in Algorithm 1.

Algorithm 1 Cross-aligned auto-encoder training. The hyper-parameters are set as $\lambda = 1, \gamma = 0.001$ and learning rate is 0.0001 for all experiments in this paper.

```
Require: Two corpora of different styles $\boldsymbol{X}_1, \boldsymbol{X}_2$. Lagrange multiplier $\lambda$, temperature $\gamma$.
  Initialize $\boldsymbol{\theta}_E, \boldsymbol{\theta}_G, \boldsymbol{\theta}_{D_1}, \boldsymbol{\theta}_{D_2}$
  repeat
    for $p = 1, 2$; $q = 2, 1$ do
      Sample a mini-batch of $k$ examples $\{\boldsymbol{x}_p^{(i)}\}_{i=1}^k$ from $\boldsymbol{X}_p$
      Get the latent content representations $\boldsymbol{z}_p^{(i)} = E(\boldsymbol{x}_p^{(i)}, \boldsymbol{y}_p)$
      Unroll $G$ from initial state $(\boldsymbol{y}_p, \boldsymbol{z}_p^{(i)})$ by feeding $\boldsymbol{x}_p^{(i)}$, and get the hidden states sequence $\boldsymbol{h}_p^{(i)}$
      Unroll $G$ from initial state $(\boldsymbol{y}_q, \boldsymbol{z}_p^{(i)})$ by feeding previous soft output distribution with temperature $\gamma$, and get the transferred hidden states sequence $\tilde{\boldsymbol{h}}_p^{(i)}$
    end for
    Compute the reconstruction $\mathcal{L}_{\text{rec}}$ by Eq. (3.2)
    Compute $D_1$'s (and symmetrically $D_2$'s) loss:
        $$\mathcal{L}_{\text{adv}_1} = -\frac{1}{k}\sum_{i=1}^{k} \log D_1(\boldsymbol{h}_1^{(i)}) - \frac{1}{k}\sum_{i=1}^{k} \log(1 - D_1(\tilde{\boldsymbol{h}}_2^{(i)})) \quad (3.8)$$
    Update $\{\boldsymbol{\theta}_E, \boldsymbol{\theta}_G\}$ by gradient descent on loss
        $$\mathcal{L}_{\text{rec}} - \lambda(\mathcal{L}_{\text{adv}_1} + \mathcal{L}_{\text{adv}_2}) \quad (3.9)$$
    Update $\boldsymbol{\theta}_{D_1}$ and $\boldsymbol{\theta}_{D_2}$ by gradient descent on loss $\mathcal{L}_{\text{adv}_1}$ and $\mathcal{L}_{\text{adv}_2}$ respectively
  until convergence
Ensure: Style transfer functions $G(\boldsymbol{y}_2, E(\cdot, \boldsymbol{y}_1)) : \mathcal{X}_1 \to \mathcal{X}_2$ and $G(\boldsymbol{y}_1, E(\cdot, \boldsymbol{y}_2)) : \mathcal{X}_2 \to \mathcal{X}_1$
```

Chapter 4

Experiments

4.1 Setup

Sentiment modification Our first experiment focuses on text rewriting with the goal of changing the underlying sentiment, which can be regarded as "style transfer" between negative and positive sentences. We run experiments on Yelp restaurant reviews, utilizing readily available user ratings associated with each review. Following standard practice, reviews with rating above three are considered positive, and those below three are considered negative. While our model operates at the sentence level, the sentiment annotations in our dataset are provided at the document level. We assume that all the sentences in a document have the same sentiment. This is clearly an oversimplification, since some sentences (e.g., background) are sentiment neutral. Given that such sentences are more common in long reviews, we filter out reviews that exceed 10 sentences. We further filter the remaining sentences by eliminating those that exceed 15 words. The resulting dataset has 250K negative sentences, and 350K positive ones. The vocabulary size is 10K after replacing words occurring less than 5 times with the "⟨unk⟩" token. As a baseline model, we compare against the control-gen model of Hu et al. [23].

To quantitatively evaluate the transfered sentences, we adopt a model-based evaluation metric similar to the one used for image transfer [24]. Specifically, we measure how often a transferred sentence has the correct sentiment according to a pre-trained sentiment classifier. For this purpose, we use the TextCNN model as described in Kim

[27]. On our simplified dataset for style transfer, it achieves nearly perfect accuracy of 97.4%.

While the quantitative evaluation provides some indication of transfer quality, it does not capture all the aspects of this generation task. Therefore, we also perform two human evaluations on 500 sentences randomly selected from the test set[1]. In the first evaluation, the judges were asked to rank generated sentences in terms of their fluency and sentiment. Fluency was rated from 1 (unreadable) to 4 (perfect), while sentiment categories were "positive", "negative", or "neither" (which could be contradictory, neutral or nonsensical). In the second evaluation, we evaluate the transfer process comparatively. The annotator was shown a source sentence and the corresponding outputs of the systems in a random order, and was asked "Which transferred sentence is semantically equivalent to the source sentence with an opposite sentiment?". They can be both satisfactory, A/B is better, or both unsatisfactory. Note that the two evaluations are not redundant. For instance, a system that always generates the same grammatically correct sentence with the right sentiment independently of the source sentence will score high in the first evaluation setup, but low in the second one.

Word substitution decipherment Our second set of experiments involves decipherment of word substitution ciphers, which has been previously explored in NLP literature [14, 43]. These ciphers replace every word in plaintext (natural language) with a cipher token according to a 1-to-1 substitution key. The decipherment task is to recover the plaintext from ciphertext. It is trivial if we have access to parallel data. However we are interested to consider a non-parallel decipherment scenario. For training, we select 200K sentences as $\boldsymbol{X}_1$, and apply a substitution cipher f on a different set of 200K sentences to get $\boldsymbol{X}_2$. While these sentences are non-parallel, they are drawn from the same distribution from the review dataset. The development and test sets have 100K parallel sentences $\boldsymbol{D}_1 = \{\boldsymbol{x}^{(1)}, \cdots, \boldsymbol{x}^{(n)}\}$ and $\boldsymbol{D}_2 = \{f(\boldsymbol{x}^{(1)}), \cdots, f(\boldsymbol{x}^{(n)})\}$. We can quantitatively compare between $\boldsymbol{D}_1$ and transferred (deciphered) $\boldsymbol{D}_2$ using Bleu score [44].

Clearly, the difficulty of this decipherment task depends on the number of substi-

[1] we eliminated 37 sentences from them that were judged as neutral by human judges.

tuted words. Therefore, we report model performance with respect to the percentage of the substituted vocabulary. Note that the transfer models do not know that f is a word substitution function. They learn it entirely from the data distribution.

In addition to having different transfer models, we introduce a simple decipherment baseline based on word frequency. Specifically, we assume that words shared between $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ do not require translation. The rest of the words are mapped based on their frequency, and ties are broken arbitrarily. Finally, to assess the difficulty of the task, we report the accuracy of a machine translation system trained on a parallel corpus [30].

Word order recovery Our final experiments focus on the word ordering task, also known as bag translation [6, 47]. By learning the style transfer functions between original English sentences $\boldsymbol{X}_1$ and shuffled English sentences $\boldsymbol{X}_2$, the model can be used to recover the original word order of a shuffled sentence (or conversely to randomly permute a sentence). The process to construct non-parallel training data and parallel testing data is the same as in the word substitution decipherment experiment. Again the transfer models do not know that f is a shuffle function and learn it completely from data.

4.2 Results

Sentiment modification Table 4.1 and Table 4.2 show the performance of various models for both human and automatic evaluation. The control-gen model of Hu et al. [23] performs better in terms of sentiment accuracy in both evaluations. This is not surprising as their generation is directly guided by a sentiment classifier. Their system also achieves higher fluency score. However, these gains do not translate into improvements in terms of the overall transfer, where our model faired better. As can be seen from the examples listed in Table 4.3, our model is more consistent with the grammatical structure and semantic meaning of the source sentence. In contrast, their model achieves sentiment change by generating an entirely new sentence which has little overlap with the source. The discrepancy between the two experiments demonstrates the crucial importance of developing appropriate evaluation measures

Method	accuracy
Hu et al. [23]	83.5
Variational auto-encoder	23.2
Aligned auto-encoder	48.3
Cross-aligned auto-encoder	78.4

Table 4.1: Sentiment accuracy of transferred sentences, as measured by a pretrained classifier.

Method	sentiment	fluency	overall transfer
Hu et al. [23]	70.8	3.2	41.0
Cross-align	62.6	2.8	41.5

Table 4.2: Human evaluations on sentiment, fluency and overall transfer quality. Fluency rating is from 1 (unreadable) to 4 (perfect). Overall transfer quality is evaluated in a comparative manner, where the judge is shown a source sentence and two transferred sentences, and decides whether they are both good, both bad, or one is better.

to compare models for style transfer.

Word substitution decipherment Table 4.4 summarizes the performance of our model and the baselines on the decipherment task, at various levels of word substitution. Consistent with our intuition, the last row in this table shows that the task is trivial when the parallel data is provided. In non-parallel case, the difficulty of the task is driven by the substitution rate. Across all the testing conditions, our cross-aligned model consistently outperforms its counterparts. The difference becomes more pronounced as the task becomes harder. When the substitution rate is 20%, all methods do a reasonably good job in recovering substitutions. However, when 100% of the words are substituted (as expected in real language decipherment), the poor performance of variational autoencoder and aligned auto-encoder rules out their application for this task.

Word order recovery The last column in Table 4.4 demonstrates the performance on the word order recovery task. Order recovery is much harder—even when trained with parallel data, the machine translation model achieves only 64.6 Bleu score. Note

From negative to positive
consistently slow . consistently good . consistently fast .
my goodness it was so gross . my husband 's steak was phenomenal . my goodness was so awesome .
it was super dry and had a weird taste to the entire slice . it was a great meal and the tacos were very kind of good . it was super flavorful and had a nice texture of the whole side .
From positive to negative
i love the ladies here ! i avoid all the time ! i hate the doctor here !
my appetizer was also very good and unique . my bf was n't too pleased with the beans . my appetizer was also very cold and not fresh whatsoever .
came here with my wife and her grandmother ! came here with my wife and hated her ! came here with my wife and her son .

Table 4.3: Sentiment transfer samples. The first line is an input sentence, the second and third lines are the generated sentences after sentiment transfer by Hu et al. [23] and our cross-aligned auto-encoder, respectively.

Method	Substitution decipher					Order recover
	20%	40%	60%	80%	100%	
No transfer (copy)	56.4	21.4	6.3	4.5	0	5.1
Unigram matching	74.3	48.1	17.8	10.7	1.2	-
Variational auto-encoder	79.8	59.6	44.6	34.4	0.9	5.3
Aligned auto-encoder	81.0	68.9	50.7	45.6	7.2	5.2
Cross-aligned auto-encoder	**83.8**	**79.1**	**74.7**	**66.1**	**57.4**	**26.1**
Parallel translation	99.0	98.9	98.2	98.5	97.2	64.6

Table 4.4: Bleu scores of word substitution decipherment and word order recovery.

that some generated orderings may be completely valid (e.g., reordering conjunctions), but the models will be penalized for producing them. In this task, only the cross-aligned auto-encoder achieves grammatical reorder to a certain extent, demonstrated by its Bleu score 26.1. Other models fail this task, doing no better than no transfer.

Chapter 5

Related work

5.1 Style transfer in vision

Non-parallel style transfer has been extensively studied in computer vision. Transferring the style of one image to the style of another can be considered a problem of texture transfer. While early methods are based on non-parametric algorithms for texture synthesis, recent deep-learning approaches have achieved high quality image style transfer results that have super resolution and are photo-realistic [17, 56, 36, 37, 50, 26, 54].

Gatys et al. [17] explicitly extract content and style representations of images from a convolutional neural network pre-trained for object recognition. Then they synthesize a new image that simultaneously matches the content representation of one image and the style representation of another.

More recent approaches learn generative networks directly via generative adversarial training [18]. Given two image domains $\mathcal{X}_1$ and $\mathcal{X}_2$, to transfer between them is to learn mappings $G : \mathcal{X}_1 \to \mathcal{X}_2$ and $F : \mathcal{X}_2 \to \mathcal{X}_1$, such that the distribution of images from $G(\boldsymbol{x}_1)$ is indistinguishable from the distribution of $\boldsymbol{x}_2$, and the distribution of $F(\boldsymbol{x}_2)$ is indistinguishable from $\boldsymbol{x}_1$. These can be formulated as adversarial losses. However, without aligned image pairs, the mapping is highly under-constrained and there are an infinite number of them, as the joint distribution is not uniquely determined by the marginal distributions. For this, CoupledGAN [36] employs weight sharing between G and F to limit the network capacity, and CycleGAN [56] introduces cycle consistency that $F(G(\boldsymbol{x}_1)) \approx \boldsymbol{x}_1$ and $G(F(\boldsymbol{x}_2)) \approx \boldsymbol{x}_2$ based on transitivity to regularize the transfer functions.

While at a high level our problem setting of language style transfer is similar to image style transfer, the discreteness of natural language does not allow us to reuse these models and necessitates the development of new methods.

5.2 Neural networks for text generation

Deep neural networks trained with large-scale corpora have reached impressive performance on text generation tasks like language modeling [3, 41], machine translation [49, 2, 53], and image captioning [40, 51, 13]. Recurrent neural networks (RNNs) are popular models to process sequential data such as sentences. They have an internal memory (i.e. hidden state) that is successively updated given the current input to capture the information about what has been calculated so far [22, 10, 11]. The hidden states efficiently encode sentences into fixed-size vector representations, based on which sequence-to-sequence (seq2seq) models [10, 49] and text auto-encoders [12, 5] are developed to learn a mapping between sentences and latent vector representations. These text generation models form the infrastructure of our method for language style transfer.

Sequence-to-sequence models Seq2seq models have enjoyed great success in a variety of tasks including machine translation [53], speech recognition [7], and text summarization [46]. They first read the source sequence using an encoder RNN to build its vector representation, which is then passed through a decoder RNN to generate the target sequence. The encoder and decoder are trained jointly and form an end-to-end seq2seq model. It can capture long-range dependencies in language and produce fluent sentences, as demonstrated by state-of-the-art systems.

Text auto-encoders A text auto-encoder uses an RNN to read an input sentence into a single vector, and then use this vector to reconstruct the original sentence [12]. Auto-encoders were originally proposed to perform data compression and dimensionality reduction [4, 20], and recently become prevalent for unsupervised learning of generative models [28, 29, 31].

Variational auto-encoders (VAEs) [28] inherit the architecture of standard auto-encoders, but make explicit assumptions about the distribution of latent representation $\boldsymbol{z}$, usually as Gaussian. They assume that the data $\boldsymbol{x}$ is generated by a directed graphical model, where the decoder learns $p(\boldsymbol{x}|\boldsymbol{z})$, and the encoder learns an approximation to the posterior distribution $p(\boldsymbol{z}|\boldsymbol{x})$. The encoder and decoder are trained to maximize the variational lower bound of data likelihood, which consists of the negative reconstruction error and a KL-divergence regularizer. Bowman et al. [5] use a VAE to generate sentences from a continuous space, and employ KL cost annealing and word dropout techniques to mitigate collapsing behavior.

Non-parallel sentence transfer Most systems for translation, summarization, and other text generation tasks are trained using parallel sentences. Our work most closely relates to approaches that do not utilize parallel data, but instead guide sentence generation from an indirect training signal [42, 23]. These approaches combine a VAE with property discriminators for controlled manipulations. For instance, Mueller et al. [42] revise the latent representation to generate sentences that better satisfy a desired property (e.g., sentiment) as measured by a corresponding classifier. However, their model does not necessarily enforce content preservation. More similar to our work, Hu et al. [23] aims at generating sentences with controllable attributes by learning disentangled latent representations [9]. They use independency constraints to enforce that attributes can be reliably inferred back from generated sentences. While our model builds on distributional cross-alignment rather than VAE for the purpose of style transfer and content preservation, these constraints can be added in the same way.

5.3 Adversarial training over discrete samples

Generative adversarial networks (GANs) GAN is a framework to train deep generative models via an adversarial process, in which two neural networks are trained simultaneously to compete with each other [18]. GAN tries to approximate the data distribution $p(\boldsymbol{x})$ by learning a generative network G that transforms a noise variable

$\boldsymbol{z} \sim p(\boldsymbol{z})$ into a sample $G(\boldsymbol{z})$. G is trained by playing against a discriminator network D, which aims to distinguish between real samples coming from the data distribution and fake samples coming from the generator distribution. The objective is a minimax game, where the Nash equilibrium is achieved when G recovers the data distribution, and D outputs probability 1/2 everywhere. In this way, GAN circumvents the intractable likelihood computations of $p_G(\boldsymbol{x})$ that would be required for explicit density models, and the entire system can be trained with backpropagation.

GANs have been successfully applied in many fields, most notably image generation, editing, and inpainting, as well as video prediction, 3D object generation, and domain adaptation.

Despite GANs' considerable success in generating real-valued data, their performance in text generation is limited where a data sample is a sequence of discrete tokens. Recently, a wide range of techniques addresses challenges associated with adversarial training over discrete samples generated by recurrent networks [55, 33, 21, 8]. In our work, we employ the Professor-Forcing algorithm [33] which was originally proposed to close the gap between teacher-forcing during training and self-feeding during testing for recurrent networks. This design fits well with our scenario of style transfer that calls for cross-alignment. By using continuous relaxation to approximate the discrete sampling process [25, 38], the training procedure can be effectively optimized through back-propagation [32, 19].

Chapter 6

Conclusions

Transferring languages from one style to another has been previously trained using parallel data. In this work, we formulate the task as *a decipherment problem* with access only to non-parallel data. The two data collections are assumed to be generated by a latent variable generative model. Through this view, our method optimizes neural networks by forcing distributional alignment (invariance) over the latent space or sentence populations. We demonstrate the effectiveness of our method on tasks that permit quantitative evaluation, such as sentiment transfer, word substitution decipherment and word ordering. The decipherment view also provides an interesting open question—*when can the joint distribution* $p(\boldsymbol{x}_1, \boldsymbol{x}_2)$ *be recovered given only marginal distributions?* We believe addressing this general question would promote the style transfer research in both vision and NLP.

Future work

Several paths of research arose from the work presented in this thesis have become active research areas:

- **Applications in transfer of various styles** [15, 35, 45] There are potentially a good deal of interesting style transfer applications, such as to transfer between professional movie reviews written by critics and unprofessional movie reviews written by general audience, to transfer a factual sentence into romantic or humorous style, to transfer between different political slant, and many more.

Moreover, many applications have real/commercial value, for instance to rewrite a tweet to increase its popularity, and to edit a news title or an advertisement to improve its click-through rate.

- **Unsupervised machine translation** [34, 1] We can think of different languages as different styles, and machine translation is to keep the meaning of a sentence and express it in another language. The success of our method in word substitution decipherment and word ordering suggests promise in unsupervised machine translation from monolingual corpora without the use of parallel sentences.

- **Evaluation metrics for style transfer** [16, 35] The evaluation of language style transfer is not easy as we must examine two aspects: whether the transferred sentence has the target style and whether it preserves the source content. When the style is subtle, such as personal writing styles, it is even hard for humans to decide how well a sentence matches with a style. It is crucial to develop appropriate evaluation metrics to systematically compare different models for advances in this area.

Bibliography

[1] Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. Unsupervised neural machine translation. *arXiv preprint arXiv:1710.11041*, 2017.

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

[3] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of machine learning research*, 3 (Feb):1137–1155, 2003.

[4] Hervé Bourlard and Yves Kamp. Auto-association by multilayer perceptrons and singular value decomposition. *Biological cybernetics*, 59(4):291–294, 1988.

[5] Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349*, 2015.

[6] Peter F Brown, John Cocke, Stephen A Della Pietra, Vincent J Della Pietra, Fredrick Jelinek, John D Lafferty, Robert L Mercer, and Paul S Roossin. A statistical approach to machine translation. *Computational linguistics*, 16(2): 79–85, 1990.

[7] William Chan, Navdeep Jaitly, Quoc V Le, and Oriol Vinyals. Listen, attend and spell. *arXiv preprint arXiv:1508.01211*, 2015.

[8] 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.

[9] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems*, 2016.

[10] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[11] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[12] Andrew M Dai and Quoc V Le. Semi-supervised sequence learning. In *Advances in Neural Information Processing Systems*, pages 3079–3087, 2015.

[13] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2625–2634, 2015.

[14] Qing Dou and Kevin Knight. Large scale decipherment for out-of-domain machine translation. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 266–275. Association for Computational Linguistics, 2012.

[15] Jessica Ficler and Yoav Goldberg. Controlling linguistic style aspects in neural language generation. *arXiv preprint arXiv:1707.02633*, 2017.

[16] Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan. Style transfer in text: Exploration and evaluation. *arXiv preprint arXiv:1711.06861*, 2017.

[17] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2414–2423, 2016.

[18] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[19] Kartik Goyal, Chris Dyer, and Taylor Berg-Kirkpatrick. Differentiable scheduled sampling for credit assignment. *arXiv preprint arXiv:1704.06970*, 2017.

[20] Geoffrey E Hinton and Richard S Zemel. Autoencoders, minimum description length and helmholtz free energy. In *Advances in neural information processing systems*, pages 3–10, 1994.

[21] R Devon Hjelm, Athul Paul Jacob, Tong Che, Kyunghyun Cho, and Yoshua Bengio. Boundary-seeking generative adversarial networks. *arXiv preprint arXiv:1702.08431*, 2017.

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

[23] Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. Controllable text generation. *arXiv preprint arXiv:1703.00955*, 2017.

[24] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *arXiv preprint arXiv:1611.07004*, 2016.

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

[26] Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jungkwon Lee, and Jiwon Kim. Learning to discover cross-domain relations with generative adversarial networks. *arXiv preprint arXiv:1703.05192*, 2017.

[27] Yoon Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

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

[29] Diederik P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, pages 3581–3589, 2014.

[30] Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M Rush. Opennmt: Open-source toolkit for neural machine translation. *arXiv preprint arXiv:1701.02810*, 2017.

[31] Tejas D Kulkarni, William F Whitney, Pushmeet Kohli, and Josh Tenenbaum. Deep convolutional inverse graphics network. In *Advances in Neural Information Processing Systems*, pages 2539–2547, 2015.

[32] 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.

[33] Alex M Lamb, Anirudh Goyal ALIAS PARTH GOYAL, Ying Zhang, Saizheng Zhang, Aaron C Courville, and Yoshua Bengio. Professor forcing: A new algorithm for training recurrent networks. In *Advances In Neural Information Processing Systems*, pages 4601–4609, 2016.

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

[35] Juncen Li, Robin Jia, He He, and Percy Liang. Delete, retrieve, generate: A simple approach to sentiment and style transfer. *arXiv preprint arXiv:1804.06437*, 2018.

[36] Ming-Yu Liu and Oncel Tuzel. Coupled generative adversarial networks. In *Advances in Neural Information Processing Systems*, pages 469–477, 2016.

[37] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. *arXiv preprint arXiv:1703.00848*, 2017.

[38] 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.

[39] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.

[40] Junhua Mao, Wei Xu, Yi Yang, Jiang Wang, Zhiheng Huang, and Alan Yuille. Deep captioning with multimodal recurrent neural networks (m-rnn). *arXiv preprint arXiv:1412.6632*, 2014.

[41] Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Interspeech*, volume 2, page 3, 2010.

[42] Jonas Mueller, Tommi Jaakkola, and David Gifford. Sequence to better sequence: continuous revision of combinatorial structures. *International Conference on Machine Learning (ICML)*, 2017.

[43] Malte Nuhn and Hermann Ney. Decipherment complexity in 1: 1 substitution ciphers. In *ACL (1)*, pages 615–621, 2013.

[44] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics, 2002.

[45] Shrimai Prabhumoye, Yulia Tsvetkov, Ruslan Salakhutdinov, and Alan W Black. Style transfer through back-translation. *arXiv preprint arXiv:1804.09000*, 2018.

[46] Alexander M Rush, Sumit Chopra, and Jason Weston. A neural attention model for abstractive sentence summarization. *arXiv preprint arXiv:1509.00685*, 2015.

[47] Allen Schmaltz, Alexander M. Rush, and Stuart Shieber. Word ordering without syntax. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2319–2324. Association for Computational Linguistics, 2016.

[48] Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. Style transfer from non-parallel text by cross-alignment. In *Advances in Neural Information Processing Systems*, pages 6833–6844, 2017.

[49] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[50] Yaniv Taigman, Adam Polyak, and Lior Wolf. Unsupervised cross-domain image generation. *arXiv preprint arXiv:1611.02200*, 2016.

[51] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164, 2015.

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

[53] 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.

[54] Zili Yi, Hao Zhang, Ping Tan Gong, et al. Dualgan: Unsupervised dual learning for image-to-image translation. *arXiv preprint arXiv:1704.02510*, 2017.

[55] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. Seqgan: sequence generative adversarial nets with policy gradient. *arXiv preprint arXiv:1609.05473*, 2016.

[56] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv preprint arXiv:1703.10593*, 2017.