ROBUST UNSUPERVISED STYLE TRANSFER ARCHITECTURE FOR COMPLEX DISCRETE STRUCTURES AND VARYING STYLES

A PREPRINT

Cairo Mo
cairomo@stanford.edu

Cherie Xu
cxuhan@stanford.edu

Alice Yang
xyang13@stanford.edu

March 19, 2019

ABSTRACT

The task of transferring style involves taking one sequence of text of a certain linguistic style and outputting the same content of text but in a given different style [1]. Recently, the proposed adversarially regularised autoencoder (ARAE) uses the latent space to generate natural outputs. The latent representation can then be trained to transfer style from unaligned text [2, 3]. In this work, we aim to improve upon Zhao et al's ARAE architecture to improve unsupervised stylistic transfer that can work robustly on longer sequences of words, such as documents and with less distinct style features such as formality. We introduce modifications upon the ARAE architecture *e.g.* hidden layer size, weight initialization and introduce a CNN based alternative for the GAN module. The efficacy of said modifications have been tested and evaluated using both automatic and human evaluation metrics

1 Introduction

Language style is a component of written text that varies commonly with the context, audience, and purpose of said text. Language style transfer is the task of imposing a target style to the content of a source sentence. In most cases, style is an abstract notion reflected in variation in word choice, sentence and paragraph structure, and punctuation that is not easily identifiable or isolated from the semantic content. Style transfer requires the disentanglement of representations of attributes *e.g.* negative/positive sentiment, plaintext/ciphertext orthography from the underlying semantic content. Breakthroughs in style transfer would have important implications for natural language processing supertasks such as natural language generation and language modeling as style is a metafeature of a language and would indicate a certain proficiency of natural language processing ability. While there are advances in style transfer in other domains such as computer vision, textual style transfer faces challenges such as the lack of parallel data and reliable evaluation metrics. Stylistic-semantic decomposition in computer vision can be achieved through convolutional neural networks due to the hierarchical invariance and abstraction of smaller visual details in higher layers of the CNN. However, textual information does not afford the same continuity observed in images, and there seems not be a characteristic scale where the style information of text is explicit. In other words, there is no specific level such as character, word, sentence, or passage level where the style can be solely observed: style occurs in any of those scales. To complicate the matter more, style is a subjective and abstract component of text that is even difficult to evaluate by human standard.

Most textual style transfer models utilizes Ad-hoc defined style classes such as sentiment [4], formality, etc. The style is enforced by a binary sentiment/style classifier trained on the corpus of different style or sentiment. The text style is called ad-hoc, since here the notion of the style is rigorously reverse-engineered out of a given training dataset. The problem set up in this form of Ad-hoc style classes are clear and the datasets are generated based on human evaluation which makes the style transfer tasks trained on this sort of tasks especially useful: style transfer success is easily measurable by the percentage of correctly classified sentences by the binary classifier. This will be the class of style transfer problem that we will tackle in this project [4].

Alternatively, style transfer task has been in the past reduced into an analogous neural machine translation (NMT) task. However, this practice has been majorly hindered by the lack of nonparallelized datasets [4]. To work around this deficient, researchers found work-arounds using generative adversarial networks (GAN) or zero-shot NMT by creating latent representation of the text that correspond separately to the style and semantic content of n some latent representations that would correspond to stylistics and semantics separately. This can be done in several ways: **a.** aligning word and sentence embedding such that the embedding state-space can be segmented into the semantic and stylistic sections, **b.** using double transfer (there-and-back) as a way of regulating the quality to the style transfer, **c.** training a stylistic discriminator as a part of the GAN. We will be taking approach c in the post NMT methods of transferring while utilizing Ad-hoc defined style classes, formality of lanaguge.

We aim to create an unsupervised stylistic transfer architecture for language style that can function robustly on varied inputs and longer sentence length. We will primarily leverage an encoder-decoder architecture for the latent representation of style and content of the sentences that will be jointly trained with a Generative Adversarial Network for the unsupervised text generation. We aim to construct a robust unsupervised style transfer architecture by combining the strength of various state-of-the-art architectures to allow the transfer source content with arbitrary styles into sentences of target style. Our generated sentences will be evaluated with metrics defined below. We hope for improvements in performance compared to other available architectures.

2 Related work

Previous attempts at language stylistic transfer relied on parallel pool of data such as sequence-to-sequence neural network models and rule-based manipulation [5]. Unlike style transfer on images where parallel data can be easily established, large scale parallel data are not available in most cases, while other methods allowed stylistic transfers on non-parallel data by reframing the stylistic transfer as a neural machine translation task [6]. Recent attempts of style transfer instead approach the task by learning a latent representation from the text to disentangle the content from the source style and recombine the latent representation of the content with some representation of the target to then generate a corresponding sentence. Unsupervised frameworks such as generative adversarial models provide the current state of the art performance for style transfer on natural language.

Our main inspiration is an architecture titled "Adversarially Regularized Autoencoder" (henceforth ARAE) by Zhao and colleagues. ARAE draws on deep latent variable models such as variational autoencoder and GAN which has shown promising result in learning smooth representation on high dimensional continuous data such as images. These latent representation allows smooth transformation in the latent space to allow complex modification of images. However, unlike image based latent representation, text sequences contain discrete structures which make the optimization of continuous latent representation difficult [2]. ARAE makes the latent representation robust on textual sequence input as it is formalized under the Wasserstein autoencoder (WAE) framework (Tolstikhin et al., 2018), which can be robustly extended to latent variable models with discrete output. Zhao and colleagues demonstrated that autoencoder cross-entropy loss upper-bounds the total variational distance between the model/data distributions. Importantly, this model is able to adapt to unaligned transfer. Unaligned transfer is the case where the model needs to change an attribute of a discrete input but does not have aligned examples, which includes tasks like changing the topic/style of a sentence [2]. ARAE architecture has shown promising results on short sentences but has short coming with more complex document. It has produced results with high sentiment transfer but relatively poor BLEU score evaluation. The paper has shown that unaligned textual transfer and textual generation can be achieved through manipulation of latent variable space derived from discrete structures. It is also question raising whether sentiment transfer can be considered style transfer as sentiment is more closely related to the semantic content of the textual sequence than the abstract sense of textual style. Our project will be addressing the issue of stylistic transfer on non-sentiment data such as language formality and register which is less closely tied with the semantic content of the sentence.

3 Approach

3.1 Architectures in Question

We compared a few approaches to style transfer on textual sequences: First, we will be treating the style transfer task as a neural machine translation task. We have implemented two different approaches of neural machine translation

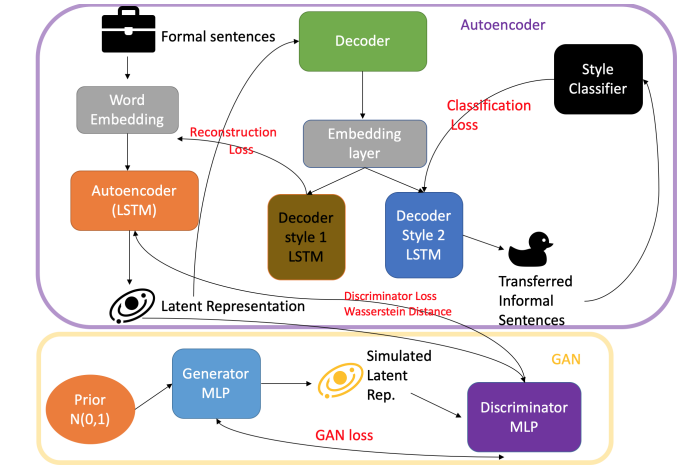


Figure 1: ARAE Architecture for Style Transfer

style transfer model based on architectures that we have previously examined in class, namely Seq2seq with GloVe embedding and Seq2Seq with character-level word embedding.

The base architecture of our model is inspired by Zhao et al. 2018 paper "Adversarially Regularized Autoencoder" [2]. We use the adversarially regularized autoencoder to create a smooth hidden encoding for discrete sequences such as text. The goal of the ARAE is to provide smoother hidden encoding for the latent representation of the style of the text and the content. This baseline architecture can be divided into three components: the training the encoder-decoder with reconstruction loss, training the critic w that discriminates between what data is real or generated and the attribute classifier that classifies whether the transferred data is of the desired target style, and lastly training the joint adversarial training of the encoder and generator from the GAN. In short, the autoencoder encoder creates a latent representation of a sentence of a given style while decoder portion of the autoencoder can decode such latent representation into a sentence of the same content of the target style. This process is regularized the presence of a GAN and style classifier to determine the ad-hoc style transfer success which will be formality in our case.

When choosing the prior distribution $\mathbb{P}_{\boldsymbol{z}}$, the choice has a large impact on the model's performance. While it may be simplest to use a fixed prior distribution, it can be too constrained. Instead, the ARAE can use a learned prior parameterised through a generator model, similar to learned priors in variational autoencoders.

More specifically, the encoder and decoder of the autoencoder utilized seq2seq model to map the text data $\mathbf{x}$ to latent space $\mathbf{z}$. For the encoder, the text is first send through an embedding layer, and then LSTM layer where $\mathbf{z}$ is the last state of the LSTM output. The decoder takes in the latent representation $\mathbf{z}$, and re-embed it in an embedding layer before sending through the embedded output through another LSTM layer. In this case, we train the two separate decoder LSTMs, one for formal style, $p(x|z, y = 1)$, and one for informal style $p(x|z, y = 0)$ and incorporate adversarial training of the encoder to remove sentiment information from the prior. The decoder outputs the conditional probability distribution $p(x|z)$ based on the specified style. The discrete distribution is estimated over the softmax of the projected latent variable $\mathbf{z}$ for each individual vocab. The reconstructive loss is calculated between the original sentence and the reconstructed sentence.

In the GAN-based regularization for the autoencoder, the latent representational output from the autoencoder and the simulated latent representation sampled from Gaussian distribution, $s \sim \mathcal{N}(0, 1)$ are passed into the discriminator. The loss from the discriminator is backpropagated through the autoencoder. The style classifier uses a MLP architecture, and the probability distribution $p_u\left(y|\mathbf{z}\right)$ is estimated for the text transfer where y represents the style label of the text. We do so by incurring a cost that is based on $p_u\left(1 - y|\mathbf{z}\right)$ by inversing the style label based on the style transfer. The decoder is also trained adversarially using this distribution. The generator, discriminator of the GAN and also the style classifier uses MLP with fully connected layers, dropouts, and nonlinearity between layers.

We formalize the full loss objective as :

$$\min_{\phi,\psi,\theta} \mathcal{L}_{\text{rec}}(\phi,\psi) + \lambda^{(1)} W\left(\mathbb{P}_Q, \mathbb{P}_{\boldsymbol{z}}\right) - \lambda^{(2)} \mathcal{L}_{\text{class}}(\phi, u)$$

where $\mathcal{L}_{\text{rec}}(\phi,\psi)$ is the reconstructive loss of the autoencoder, W is the Wasserstein distance between $\mathbb{P}_Q$ and $\mathbb{P}_{\boldsymbol{z}}$ which are the distribution from a discrete encoder model, that is, $enc_\phi(\boldsymbol{x})$ and $\mathbb{P}_{\boldsymbol{z}}$ is the prior distribution for GAN respectively. $\mathcal{L}$ is the classification loss for the transferred sentences.

3.2 Original Components

While understanding the ARAE architecture, we noted that the Generator and Discriminator which regulates the autoencoder training is created based on a Multilayer Perceptron model. We hoped to improve upon these architecutre as GAN play an important role in regulating the training of the autoeconoder and allowing style transfer. Instead of MLP architecture, we were able to isolate the GANs module of the ARAE and replaced it with our original component involving two 1-dimensional convolutional neural networks for the encoder and decoder. The motivation behind using a CNN architecture is to optimize the generation and discrimination of latent space in the GAN so that it can best match the actual probability distribution. Most of the recent applications of GANs involve modeling images, while we are using a GAN as part of the architecture to model language. In Piotr Bojanowski's 2017 paper on optimizing the latent space of generative networks, he shows that autoencoders where both the encoder and decoder are deep convnets can be as successful as GANs but require less complicated training [7]. Another advantage of convolutional model is that fewer trainable parameters are needed and long range dependencies in the data can be uncovered with properly selected filter sizes. We decided to incorporate a CNN to apply the framework of having both the generator and discriminator parametrized as deep convnets to attempt to improve upon the existing ARAE architecture, extending the combination of deep convnets beyond autoencoders. Additionally, we expect the CNNs will be able to learn local features on top of the word embeddings, as described in Yoon Kim's work with CNNs for sentence classification [8]. The Gaussain prior which is sampled to generate the fake latent variable space now passes through 3 convolutional layers. To complement the gaussian nature of the data, we have also altered the activation function from a leaky Relu to a more robust Gaussian Error linear unit (GeLu) which helped with the problem of vanishing gradient and signal loss [9]. The specific hyperparameters for our models are given in Appendix B.

4 Experimentation

4.1 Dataset

We trained and tested our model on Yahoo Answers for Formality Corpus, a non-sentimental dataset with distinctive vernacular and syntax.

The GYAFC dataset is the largest dataset for one specific stylistic transfer, and consists of a corpus of informal/formal sentence pairs. The data come from Yahoo Answers, which are a large source of informal sentences. For each informal sentence in the corpus, there are 4 human-generated formal counterparts to the sentence, so this dataset provides a large corpus of aligned data that can be used for automatic evaluation metrics like BLEU. The GYAFC dataset fits the qualification of ad-hoc defined style classes as the style is enforced by a binary classifier and human oracle at the creation of the dataset. The notion of style or formality is reverse engineered based on Yahoo Answers using Amazon Mechanical Turk [1]. We have specifically selected the domain, Family Relations.

4.2 Evaluation Metrics

Altogether, we used a combination of human evaluation, still considered to be the gold standard, as well as automatic evaluation strategies including BLEU score and a classifier score.

4.2.1 Human Evaluation

In human evaluation, we will focus on three standards taken from Singh & Palod's 2018 paper and commonly used for style transfer task evaluation [3]:

1. Soundness (textual entailment between input and generated texts)

2. Coherence (comprehensibility, lack of grammatical errors)
3. Effectiveness (similarity of style of the generated texts and the desired style)

We created a Google form with a random sample of 10 generated sentences (5 from each style) for each model. Human responders evaluated each of these sentences and rated them on a scale from 1-5 on the axes of coherence, formality (our chosen style), and soundness. Formality is a measure of transfer strength, that is, how successfully the generated text embodies the target style. Additionally, for evaluating soundness of content, we are trying to evaluate whether the semantic meaning of the generated sentences can remain the same despite the style changing. Soundness acts as a measure of content preservation, since the goal is to change the style of a sentence without altering its underlying meaning. [10]

4.2.2 Machine Evaluation

In machine evaluation, we will evaluate

1. Soundness of content by using a case-insensitive BLEU metric.
2. Perplexity based on the language model
3. Effectiveness of style transfer by using a pre-trained style classifier
4. Style Transfer Effectiveness with respect to a style classifier

We will not address coherence in our machine evaluation, deferring to the standard of using human evaluation for this metric. We can use our test set to find a final percentage accuracy for machine evaluation of soundness and of effectiveness. [6]

We adjusted evaluation metrics for different cases of transfer - for example focusing more on human or machine evaluation in machine translation due to differences in level of sentiment transfer (e.g. - Positive/ Negative, formal/informal) and for non-sentimental style transfer (e.g - colloquial English to literary Shakespearean jargon) due to difficulties such as transference of meaning between different styles.

BLEU is a common automatic evaluation metric used for machine translation tasks. In the case of measuring style transfer, we can use BLEU to compare the generated text in the target style to several "gold" reference style examples. This is only possible with datasets like the GYAFC which include 4 informal references for each original formal sentence, and vice versa. Perplexity can be used for style transfer tasks by more narrowly defining what words belong or do not belong in a certain style. A lower perplexity would indicate less randomness, and thus a more specific definition of what the target style is.

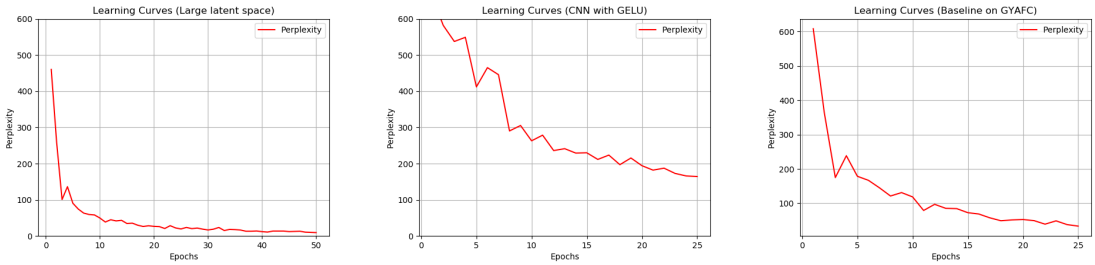
The effective of style transfer is measured by a style classifier co-trained with the training data to classify the latent representation of a formal sentences to an in formal sentence. The accuracy as outputed by this style classifier when applied to transferred sentece give us a metric of how effective the style transfer has taken place. (See Figure 3).

4.3 Experimental details

The first experiment involved changing the encoder to use a 1 dimensional convolutional neural network. The configuration of the Conv model in this case involves swapping out the multilayer perceptrons (MLP) instead for CNNs in the encoder and decoder. The motivation for this experiment is to observe whether a convolutional neural network would be able to better isolate features of linguistics style. In this experiment, we tested the Conv model compared to the MLP model.

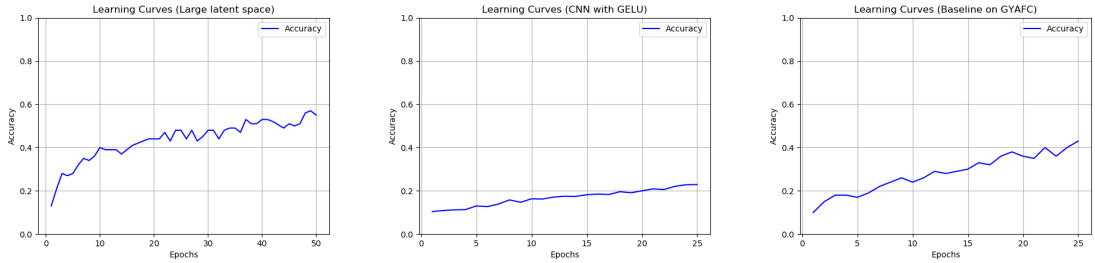
Since the baseline and Conv 1D model did not seem to be learning adequately after 25 epochs, we decided to perform an experiment to increase the size of the latent space, which is where the model performs manipulations to change the output. The motivation for this experiment is that having a larger latent space will allow the model to perform more complex manipulations to the output. By increasing the latent space dimension to 256 (up from 128), and training for 50 epochs, we observed that it was able to continue learning until the 50th epoch and continue reducing the perplexity. Compared to the baseline model run on the GYAFC dataset, the large latent model experiment achieves a lower perplexity and higher accuracy:

The experiment of increasing the latent space was able to reduce perplexity, but still did not achieve as high of an accuracy as the baseline model was able to on the Yelp data. Additionally, the generated output could be improved, as we will discuss in the Analysis section.



(a) Perplexity of large latent space (b) Perplexity of CNN + GELU model (c) Perplexity of baseline model

Figure 2: Comparison of perplexity of three models during ARAE training



(a) Accuracy of large latent space (b) Accuracy of CNN + GELU model (c) Accuracy of baseline model

Figure 3: Comparison of accuracy of three models during ARAE training

Thus, the next experiment involved incorporating GloVe embeddings [11]. The motivation behind this experiment was that incorporating pre-trained data could improve the learning of semantic. In previous experiments, generated output seemed to show a change in the semantic meaning, not just in style. We hope that incorporating a pretrained embedding could help to better disentangle style and semantics at the decoding phase.This will be discussed further in the Analysis section. GloVe embeddings first introduced as fixed word embeddings used in the autoencoder and then as weight initialization for the encoder layer that is trainable by the data. The performance of the results will be discussed in later section of the paper.

The next experiment was to test a leaky rectified linear unit (ReLU) versus a Gaussian Error Linear Unit (GELU) activation function. To testing, we subsampled a 10,000 sentence subset of our dataset (for experimentation, not accuracy, full model currently training). The original discriminator uses a MLP model with a leaky ReLU activation. In our experiment, we tested multiple architectures swapping the model and activation to use the Conv model from the first experiment in combination with the GELU activation. In total, we tried 4 different combinations of a CNN with GELU activation, MLP with GELU activation, CNN with ReLU, and the original MLP with ReLU. We also experimented

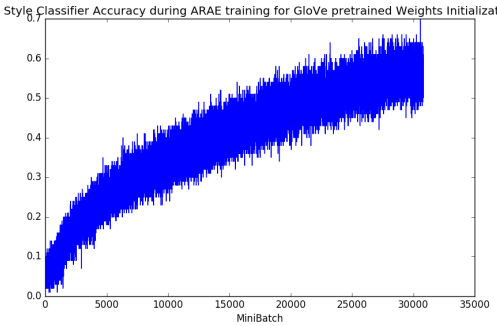


Figure 4: Training Style Classification Accuracy with GloVe Weight initialization

Automatic Evaluation*			
	Baseline on GYAFC	Large Latent Space	CNN and GELU (10k)
Corpus BLEU Decoder 1 Source	84.9	72.4	68.5
Corpus BLEU Decoder 1 Target	1.02	2.32	1.66
Corpus BLEU Decoder 2 Source	67.3	64.9	60.8
Corpus BLEU Decoder 2 Target	3.04	2.08	2.05
Accuracy	0.703	0.773	0.229
Perplexity	18.53	9.65	164.3

Table 1: **Style transfer automatic evaluation**: Evaluation of the experimentalal models on the 10k subsampled dataset.

Human Evaluation			
	Baseline on GYAFC	Large Latent Space	CNN and GELU (10k)
Soundness	1.771	2.036	2.010
Decoder 1 Coherence	3.365	2.900	4.670
Decoder 2 Coherence	3.482	2.612	2.670
Decoder 1 Formality	2.841	2.771	3.450
Decoder 2 Formality	2.329	1.894	2.560

Table 2: Style transfer evaluation by humans. Soundness is how similar the two sentences of different styles were rated, coherence is how understandable and grammatical the sentences were, and formality is a measure of effectiveness.

with alternating Conv and RELU and changing the Conv architecture size to 3-3-3. The training curves for these can be found in Appendix B.

Additionally, we performed experiments to compare the task of style transfer and machine translation. Using a word-based neural machine translation model using RNNs, we treated the source style as the source language and the target style as the target language. Additionally, we experimented with the character-based convolutional encoder for neural machine translation. Because there is less aligned data in the GYAFC dataset than for English/Spanish translations, it took around 4-6 hours to train these models.

4.4 Results

As seen in Table 1, we evaluated the baseline ARAE architecture on Yelp data, the baseline ARAE on GYAFC data, and large latent space experiment on GYAFC data, and word-based RNN NMT and character-based convolutional NMT on GYAFC data. We trained the models with two separate decoders, Decoder 1 for formal style $p(\boldsymbol{x}|\boldsymbol{z}, y = 1)$ and Decoder 2 for informal style $p(\boldsymbol{x}|\boldsymbol{z}, y = 2)$.

In Table 2, we have results in terms of soundness, coherence, and effectiveness. For each model (baseline and large latent space) ran on the GYAFC dataset, we randomly sampled 5 generated sentences from each decoder. There were 5 sentences from Decoder 1 (formal) and 5 sentences from Decoder 2 (informal), each from the two models, for a total of 20 sentences. We crowdsourced responses from a Google form sent to 18-22 year-old college students at Stanford and to individuals in this group's social networks, and received 35 responses. We asked responders to evaluate each sentence for coherence on a scale from 1-5, 5 being the most coherent. To gauge soundness, we showed responders two sentences (1 formal, 1 informal) from the same model, and asked them to rate how similar in meaning the two sentences were, disregarding style. To gauge effectiveness, we asked responders to rate sentences for formality, one a scale from 1-5, 5 being the most formal. In the case of Decoder 1, a high formality score of 5 indicates that the model generated sentences in the correct formal style. In the case of Decoder 2, a *low* formality score of 1 shows the model generated sentences in the correct *informal* style. Randomly sampled sentences are shown in Appendix A.

After this experimentation, we trained our final 3-layer convolutional model with GELU activation on the full Yahoo Formality dataset. Over 15 epochs, our accuracy increased almost consistently from .159 to 0.643 and our perplexity dropped from 296.13 to 9.065. Due to slowness of training, we were not able to train for the entire targeted 25 or 50 epochs, but expect higher accuracy, given rate of growth.

5 Analysis

A notable feature of the baseline model is that it was only able to perform well on short sentences which is to be expected as the dataset that the baseline model was tuned for are based on shorter, sentiment based sentences extracted base on Yelp Reviews [2]. Another concern that was raised in the reduced performance of the baseline model on the GYAFC is that unlike sentiment, formality is less closely related to the semantic content of the sentence. By transferring sentiment from positive to negative, semantic content of the sentence as to be altered as words can be replaced with their antonyms and sentences can be logically negated with the inclusion of words such as "not." As we have seen previously in class, a word-vector representation of the word can be clearly manipulated to reflect its antonym. This operation becomes less clear when we walk about formality of a word or a sentence's embedding.

We began to see improvement in performance once we increase the hidden latent variable space as the sentences from GYAFC dataset are generally longer and more complex compared to the Yelp dataset used in the ARAE paper. Additionally, the CNN and GELU model is rated highly in terms of coherence for formal styles, but less so for informal styles. A feature of the large latent space variation of the model is that it can generate longer sentences with higher coherence. The most coherent sentence (rated by humans) generated in the formal style by the large latent space model had a score of 4.121, and this sentence was *"i always gave up and i am still able to meet my boyfriend."* with a length of 13 words. In comparison, the baseline model's most coherent formal sentence had a score of 4.938 and this sentence was *"i am talking about him."*, which is only 5 words long.

It is also not the case that the longer the sentence the less coherent the generated text becomes. Looking at the large latent space model's least coherent sentence with a rating of 1.061, *"you can do not have it only anyone."*, it is has 8 words and is shorter than its most coherent sentence. Compared to the baseline model, however, the baseline's least coherent formal sentence is also the longest sentence it generated – *"when you are in love with him if you do not like her then you will find out if you are in love with him."* – with a score of 2.281 and length of 25 words.

Although the large latent space model did not perform as well as the baseline on automatic evaluation metrics, scoring lower on BLEU on 3 out of 4 areas and having a slightly lower accuracy, the large latent space model performs better in terms of achieving informal style transfer and soundness. Additionally, the large latent space model achieves a lower perplexity in the automatic evaluation. The structure of this model may have allowed for a better soundness score in part due to the nature of style transfer as a task. Unlike machine translation or sentiment transfer, which are tasks that change language at a word or phrasal level, style transfer is a suprasegmental feature that sits on top of the syntactic and semantic features of a given chunk of language. Hence, the task of separating out the "style" of a piece of text is not as isolated as the task of separating out the sentiment of a word from the larger phrase. The larger latent space allows for more complex transformations on the output while still remaining in the data space/manifold, and it is possible that the larger dimensions facilitate a more flexible transformation of the output text.

Another reason that the large latent space model did not perform well in terms of coherence is the presence of many `<unk>` tokens. Due to the nature of the Yahoo Answers dataset and the nature of an "informal" style, the vocabulary included many misspellings of the same word, strange entries like "..." and "....." that were disambiguated. When the vocabulary was trimmed, many of these rare vocabulary entries were left out. Some human feedback mentioned the lack of punctuation in the model's generated text, which could also have resulted from the dataset. One way to have addressed this issue would be to implement a spellchecker using edit distance to calculate the nearest word to an `<unk>` token.

With respect to the results when using GloVe as weight initialization for word embedding, while the training transfer accuracy is steadily increasing (Figure 4), the testing style transfer accuracy stayed relatively stagnant with final style transfer accuracy of .194 at epoch 25. The trainable weight initialization of GloVe seems to be significantly overfitting on the train data. The pretrained initialization seems to be suboptimal for our use case. We also ran into problem with large number of `<unk>` as our vocabulary, especially that of the informal classes did not conform with that of the Glove Embeddings. The random, low variance initliazation for parameters yielded higher accuracy in terms of style transfer

To summarize, the main challenges that we have faced can be due to symbolic information loss during the encoding, decoding, and the generation of the sentences. These challenges are not unlike the ones that previous architectures using LSTM and CNN for text generation [12]. One problem with applying GAN to text is that the

gradients from the discriminator cannot effectively back-propagate through discrete variables. We attempted to combat this problem with the inclusion of GeLu unit.

6 Conclusion

Through our experimentation, we found that the ARAE architecture which is trained for sentiment transfer does not effectively generalize to the transfer of other type of language style such as formality. In an attempt to make this model more robust and generalizable in longer textual sequences with less distinct style features than sentiment, we suggest the use of a larger latent space, and incorporating GloVe embeddings, to improve upon the adversarially regularized autoencoder (ARAE) architecture for style transfer.

The addition of a latent space with higher dimensionality, as well as the use of pre-trained data, has the potential to improve upon style transfer by achieving more style transfer without changing the semantic meaning, as well as achieving a wider range of styles including the "informal" style. The limitations of our work, and for style transfer in general, include the lack of aligned data and the vague, ill-defined nature of linguistic style. Our model was not well-equipped to handle `<unk>` tokens, and lacked sufficient data which may have allowed us to produce more coherent text in different styles.

For future work in this area, it would be worth exploring the combination of rule-based and statistical approaches, incorporating other NLP tasks such as a spell checker, or handling `<unk>` tokens by using a character-based embedding. Our model in its current state does not learn explicitly a latent representation of the language style. In the future, it would be interesting to compare methods that deliberately extract a style embedding the sentence to our model that does not explicitly extract this information but instead utilizes the style decoder as a way of transferring style.

7 Additional Information

Mentor: Chris Manning

References

[1] Tetreault Rao. Dear sir or madam, may i introduce the gyafc dataset: Corpus, benchmarks and metrics for formality style transfer. 2018. https://arxiv.org/pdf/1803.06535.pdf.

[2] Zhao, Kim, Zhang, Rush, LeCun. Adversarially regularized autoencoders. 2018. https://arxiv.org/pdf/1706.04223.pdf.

[3] Singh, Palod. Sentiment transfer using seq2seq adversarial autoencoders. 2018. https://arxiv.org/pdf/1804.04003.pdf.

[4] Alexey Tikhonov and Ivan P. Yamshchikov. What is wrong with style transfer for texts? *https://arxiv.org/pdf/1808.04365.pdf*, 2018.

[5] Jhamtani, Gangal, Hovy, Nyberg. Shakespearizing modern language using copy-enriched sequence-to-sequence models. 2017. https://arxiv.org/pdf/1707.01161.pdf.

[6] Zhang, Ren, Liu, Wang, Chen, Li, Zhou, Chen. Style transfer as unsupervised machine translation. 2018. https://arxiv.org/pdf/1808.07894.pdf.

[7] David Lopez-Paz Arthur Szlam Piotr Bojanowski, Armand Joulin. Optimizing the latent space of generative networks. *https://arxiv.org/pdf/1707.05776.pdf*, 2017.

[8] Yoon Kim. Convolutional neural networks for sentence classification. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014. https://www.aclweb.org/anthology/D14-1181.

[9] Kevin Gimpel Dan Hendrycks. Gaussian error linear unit. *https://arxiv.org/pdf/1606.08415.pdf*, 2018.

[10] Nanyun Peng-Dongyan Zhao Rui Yan Zhenxin Fu, Xiaoye Tan. Style transfer in text: Exploration and evaluation. *https://arxiv.org/pdf/1711.06861.pdf*, 2017.

[11] Christopher D. Manning Jeffrey Pennington, Richard Socher. Glove: Global vectors for word representation. *https://nlp.stanford.edu/pubs/glove.pdf*, 2014.

[12] Soujanya Poria Erik Cambria Tom Young, Devamanyu Hazarika. Recent trends in deep learning based natural language processing. *https://arxiv.org/pdf/1708.02709.pdf?*, 2018.

Appendix A

A GYAFC Style Transfer Sentences

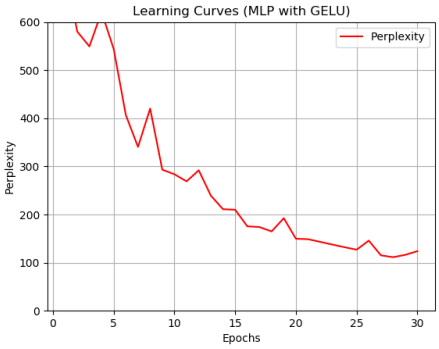
Baseline Model-generated	
Noise to Informal	Noise to Formal
when you are in love with him if you do not like her then you will find out if you are in love with him.	well if you are ready and you will be happy then you will get out with him and you are going to be happy.
yes, if you are still interested in.	yes yes you are so much fun.
that depends on their own way.	its always fun of their own way.
if you are not interested then then yes.	but you are not as as you are.
i am talking about him.	go out with her all you like.

Large Latent Space Model-generated	
Noise to Informal	Noise to Formal
you can do not have it only anyone.	you can do it only for it as long time.
you should tell him the truth of a boyfriend.	so get him back on a man!
even a friend may not go out with him, and it's not working out myself.	sounds like trying to never give him some one and talk about it
i always gave up and i am still able to meet my boyfriend.	i always found it and i am willing to work for my bf
i feel that if i was a man who has the most romantic feelings for you.	i don't really know that my guy might be the kid of my dreams is to me.

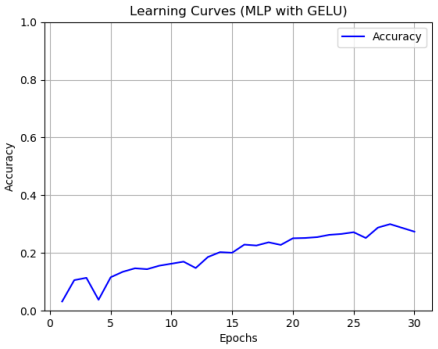
Appendix B

B Experimental Details

Experiment combining MLP with GELU:

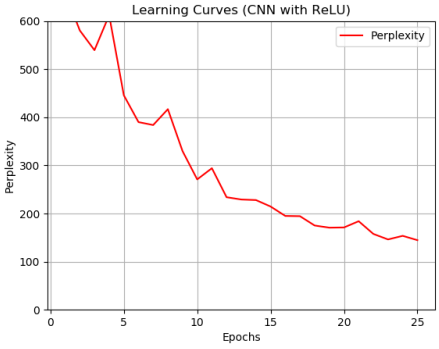


(a) Perplexity of MLP and GELU model

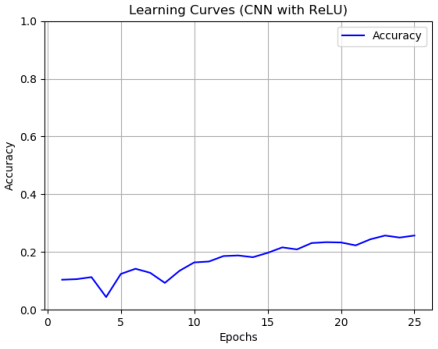


(b) Accuracy of MLP and GELU model

Experiment combining CNN with leaky ReLU:



(a) Perplexity of CNN and ReLU model



(b) Accuracy of CNN and ReLU model

Autoencoder: Seq2Seq2Decoder had embedding size of 30004, with 256 hidden layers:

- Embedding decoder 1 and decoder 2 both have embedding size of 30004, with 256 hidden layers
- Encoder is an LSTM with 256 hidden layers
- Decoder 1 and Decoder 2 are LSTMs with 256 hidden layers

GAN generator: MLP for the generator:

- Uses ReLU activation between each linear layer
- The first linear layer has 32 input features and 256 output features
- The second and third linear layers have 256 input features and 256 output features

GAN discriminator: The CNN for discriminator

- Uses a three 1-dimensional convnets with 256 hidden layers
- GeLU uses dropout with $p = 0.3$
- Uses a linear layer at the end with 7936 input features and 1 output feature

Classifier: The MLP for classifier

- Uses a three linear layers with 256 input features and the first two have 256 output features
- Uses ReLU bewteen the layers

Robust Unsupervised Style Transfer Architecture for Complex Discrete Structures

Project Addendum

Cairo Mo, Cherie Xu, Alice Yang

Below is an addendum to our final project report. At the time of our submission, we have a few models that have not finished training but have completed in time for the presentation. The results are promising in terms of improvement in automatic evaluation. We thought it would be interesting to share these further findings.

Abstract

The task of transferring style involves taking one sequence of text of a certain linguistic style and outputting the same content of text but in a given different style . Recently, the proposed adversarially regularised auto-encoder (ARAE) uses the latent space to generate natural outputs. The latent representation can then be trained to transfer style from unaligned text . In this work, we aim to improve upon Zhao et al's ARAE architecture to improve unsupervised stylistic transfer that can work robustly on longer sequences of words, such as documents and with less distinct style features such as formality. We introduce modifications upon the ARAE architecture *e.g.* hidden layer size, weight initialization and introduce a CNN based alternative for the GAN module. The efficacy of said modifications have been tested and evaluated using both automatic and human evaluation metrics

1 Model Description

1.1 Convolutional Networks for Discriminator in ARAE

This is a component of the architecture that is mentioned in our full report. However, we were only able to test this model on a subsample of 10k data points which proves to be not enough signal for the GAN to generate varying sentences.

This model is an update on the ARAE model as we replaced the discriminator in the GAN portion of the ARAE which uses two layers MLP to discriminate simulated data versus real data with a CNN based architecture. The GAN discriminator acts like a regulariser and adversarially trains the Autoencoder's encoder by minimizing the loss function $-\frac{1}{m}\sum_{i=1}^{m}\log p_u\left(1-y^{(i)}|\mathbf{z}^{(i)}\right)$, where $\mathbf{z}$ is encoded latent representation and $y^{(i)}$ is the class label of the sentence. The CNN architecture is as follows:

```
GAN Discriminator:
CNN_D(
  (conv1): Conv1d(1, 256, kernel_size=(3,), stride=(1,))
  (bn1): BatchNorm1d(254, eps=1e-05, momentum=0.1, affine=True)
  (conv2): Conv1d(256, 256, kernel_size=(5,), stride=(2,), padding=(1,))
  (bn2): BatchNorm1d(126, eps=1e-05, momentum=0.1, affine=True)
  (conv3): Conv1d(256, 1, kernel_size=(3,), stride=(1,))
  (bn3): BatchNorm1d(124, eps=1e-05, momentum=0.1, affine=True)
  (gelu): GeLU()
  (fc): Linear(in_features=7936, out_features=1, bias=True)
)
```

We first feed the latent representation of the real sentences and the latent representation generated from the Gaussian prior into the convolutional layera with kernel size 3, 5, and 3 respectively. Between layers, we employ batch normalization. One update upon the previous model is that we utilizes a GeLu activation function (Gaussian Error Linear Error Unit) in between unit. This model is trained on the full dataset for 25 epoches.

1.2 Convolutional Networks + Fully Connected Layer for Discriminator in ARAE

This Discriminator bears striking similarity to the aforementioned CNN-Discriminator. One modification has been made to this model, at the end of all convolution layers, we concatenate the input into the discriminator with the convolutional output, in an effort to preserve the syntactical information contained within the latent representation which can be lost during the convolution operations. The model summary is as follows:

```
GAN Discriminator:
CNN_D(
  (conv1): Conv1d(1, 256, kernel_size=(3,), stride=(1,), padding=(1,))
  (bn1): BatchNorm1d(256, eps=1e-05, momentum=0.1, affine=True)
  (conv2): Conv1d(256, 256, kernel_size=(5,), stride=(1,), padding=(2,))
  (bn2): BatchNorm1d(256, eps=1e-05, momentum=0.1, affine=True)
  (conv3): Conv1d(256, 1, kernel_size=(3,), stride=(1,), padding=(1,))
  (bn3): BatchNorm1d(256, eps=1e-05, momentum=0.1, affine=True)
  (gelu): GELU()
  (fc): Linear(in_features=32768, out_features=1, bias=True)
)
```

Note the last fully connected layer is much larger than the previously mentioned model as we concatenate the input to the convolutional output.

Automatic Evaluation*				
	Baseline	Large Latent Space	CNN & GELU	CNN & GELU & FC
Accuracy	0.423	0.572	0.697	0.698
Perplexity	18.53	9.65	6.79	6.66

Table 1: **Style Transfer Automatic Evaluation**: Style transfer test accuracy and perplexity of model trained on the full dataset

2 Results

See Table 1 for the breakdown of automatic evaluation metrics on the Full GYAFC Family Relations dataset in terms of style transfer success rate and style classification perplexity. Perplexity here is defined as the exponential cross entropy loss of the style classification.

Both Convnet based models showed significant improvement from the Large Latent variation of the ARAE. Both has shown higher success rate in Style Transfer. The CNN-GAN models seem to perform almost as well as the ARAE model on the sentiment transfer task (success rate 73.3%). However, the sentiment corpus that the model was trained on was on average shorter than the formality corpus sentences. Given that formality is a harder to disentangle form of language style *prima facie*, we were happy to have make the ARAE architecture more robust on more complex style passages.

In terms of Perplexity and Style Transfer accuracy, we see a steady improvement across epochs. Although we had to terminate the experiment at 25 epochs due to time constraint, it seems that the model is still improving.

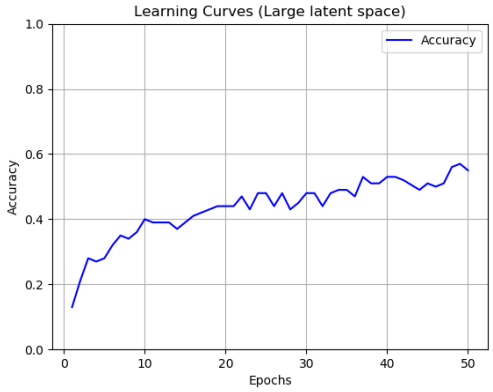
3 Analysis

The main distinguishing features between the two CNN discriminator, even though the two have performed similarly on style transfer accuracy, is the quality of the generated/transferred sentences. The following sentences are randomly selected from the Style transfer output:

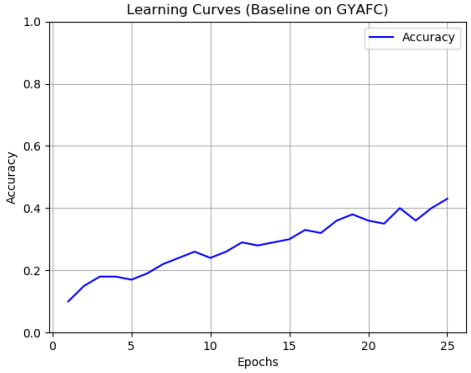
Comparison of CNN based Discriminator		
IF ? F	Without FC	With Fully connected layer
IF to F	they know the best people will like me or having fun guys be aware of you are.	tell him that he needs to understand why, you like him and many more that he likes you, but that you are more likely <unk>
F to IF	but i mean, i dont know IF you have the answer and i can keep everything i don't know of you.	when you are married, it is worth the wait.
IF to FF	i just make that age to do the answer or the real deal , why not love her last day	you should make sure alot of <unk> maybe he just gets that from the world

It seems that with CNN with fully connected layer was able to generate longer sentences that have higher grammatically upon examination. With the addition of the fully connected layer, the sentences have became more syntactically coherent, closer to minimal pairs of actual grammatical sentences. It is interesting to see convolution of on the latent representation of a sentence is able to affect the syntax of the sentence. One of our worries for applying convolutions is the interpretability of the latent representation and whether if there is any spacial mapping within the latent representation where the convolutions are able to extract long range dependencies with respect to the spacial structures. With our experiments, we found that it is necessary to have some way to regularize for the original structure of the sentence. Without the fully connected layer as a way of regularizing for syntax, a lot generated sentences appeared to be unordered small fragments of sentences that did make sense independently, but the whole sentence is ungrammatical.

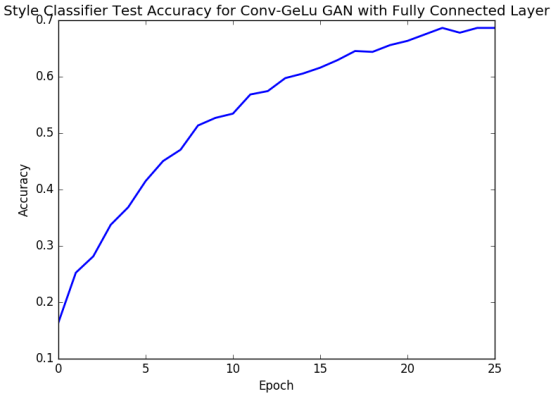
Overall, it seems the addition of the convolutional discriminator did make the style transfer more robust and better generalized to a more complex set of dataset and notion of style.



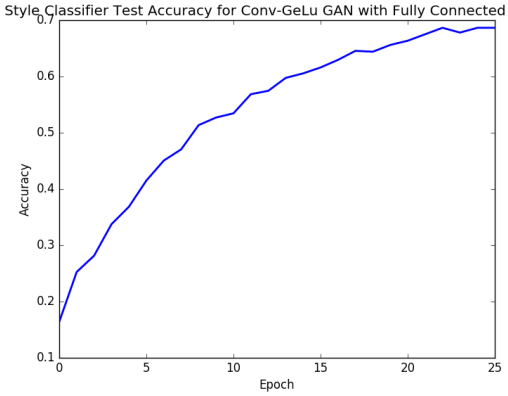
(a) Accuracy of large latent space



(b) Accuracy of Baseline ARAE

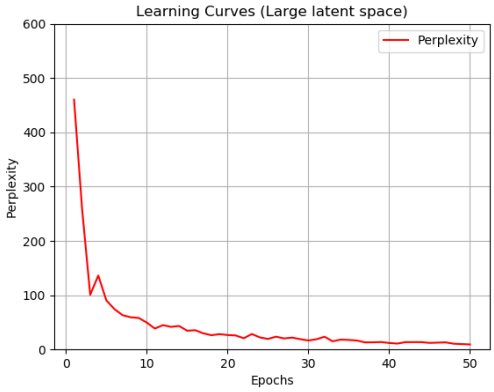


(c) Accuracy of CNN + GELU model

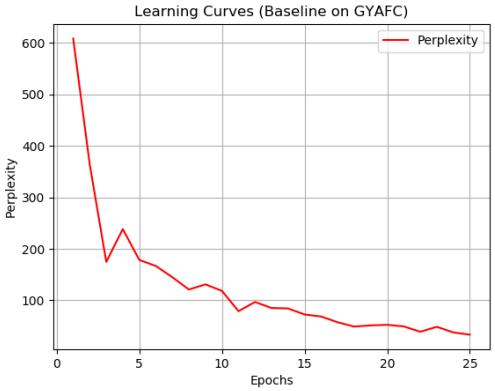


(d) CNN +GELU+ Fully connected model

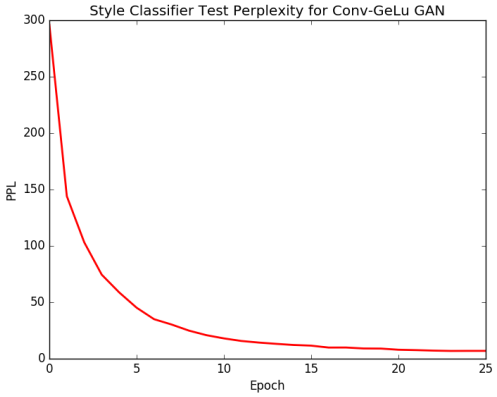
Figure 1: Comparison of Accuracy of Three Models during ARAE Training



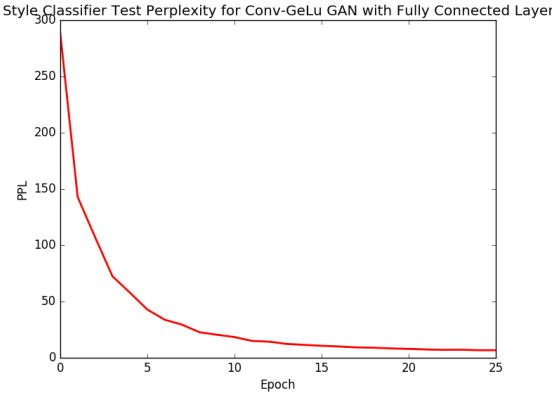
(a) Perplexity of large latent space



(b) Perplexity of baseline model



(c) Perplexity of CNN + GELU model



(d) Perplexity of CNN + GELU + FC model

Figure 2: Comparison of perplexity of three models during ARAE training