Unsupervised Evaluation Metrics and Learning Criteria for Non-Parallel Textual Transfer

Richard Yuanzhe Pang[1§] **Kevin Gimpel**[2]
[1]New York University, New York, NY 10011, USA
[2]Toyota Technological Institute at Chicago, Chicago, IL 60637, USA
yzpang@nyu.edu, kgimpel@ttic.edu

Abstract

We consider the problem of automatically generating textual paraphrases with modified attributes or properties, focusing on the setting without parallel data (Hu et al., 2017; Shen et al., 2017). This setting poses challenges for evaluation. We show that the metric of post-transfer classification accuracy is insufficient on its own, and propose additional metrics based on semantic preservation and fluency as well as a way to combine them into a single overall score. We contribute new loss functions and training strategies to address the different metrics. Semantic preservation is addressed by adding a cyclic consistency loss and a loss based on paraphrase pairs, while fluency is improved by integrating losses based on style-specific language models. We experiment with a Yelp sentiment dataset and a new literature dataset that we propose, using multiple models that extend prior work (Shen et al., 2017). We demonstrate that our metrics correlate well with human judgments, at both the sentence-level and system-level. Automatic and manual evaluation also show large improvements over the baseline method of Shen et al. (2017). We hope that our proposed metrics can speed up system development for new textual transfer tasks while also encouraging the community to address our three complementary aspects of transfer quality.

1 Introduction

We consider **textual transfer**, which we define as the capability of generating textual paraphrases with modified attributes or stylistic properties, such as politeness (Sennrich et al., 2016a), sentiment (Hu et al., 2017; Shen et al., 2017), and formality (Rao and Tetreault, 2018). An effective transfer system could benefit a range of user-facing text generation applications such as dialogue (Ritter et al., 2011) and writing assistance (Heidorn, 2000). It can also improve NLP systems via data augmentation and domain adaptation.

However, one factor that makes textual transfer difficult is the lack of parallel corpora. Advances have been made in developing transfer methods that do not require parallel corpora (see Section 2), but issues remain with automatic evaluation metrics. Li et al. (2018) used crowdsourcing to obtain manually-written references and used BLEU (Papineni et al., 2002) to evaluate sentiment transfer. However, this approach is costly and difficult to scale for arbitrary textual transfer tasks.

Researchers have thus turned to *unsupervised* evaluation metrics that do not require references. The most widely-used unsupervised evaluation uses a pretrained style classifier and computes the fraction of times the classifier was convinced of transferred style (Shen et al., 2017). However, relying solely on this metric leads to models that completely distort the semantic content of the input sentence. Table 1 illustrates this tendency.

We address this deficiency by identifying two competing goals: preserving semantic content and producing fluent output. We contribute two corresponding metrics. Since the metrics are unsupervised, they can be used directly for tuning and model selection, even on test data. The three metric categories are complementary and help us avoid degenerate behavior in model selection. For particular applications, practitioners can choose the appropriate combination of our metrics to achieve the desired balance among transfer, semantic preservation, and fluency. It is often useful to summarize the three metrics into one number, which we discuss in Section 3.3.

We also add learning criteria to the framework of Shen et al. (2017) to accord with our new metrics. We encourage semantic preserva-

[§] Work completed while the author was a student at the University of Chicago and a visiting student at Toyota Technological Institute at Chicago.

Proceedings of the 3rd Workshop on Neural Generation and Translation (WNGT 2019), pages 138–147
Hong Kong, China, November 4, 2019.
www.aclweb.org/anthology/D19-56%2d

tion by adding a "cyclic consistency" loss (to ensure that transfer is reversible) and a loss based on paraphrase pairs (to show the model examples of content-preserving transformations). To encourage fluent outputs, we add losses based on pretrained corpus-specific language models. We also experiment with multiple, complementary discriminators and find that they improve the trade-off between post-transfer accuracy and semantic preservation.

To demonstrate the effectiveness of our metrics, we experiment with textual transfer models discussed above, using both their Yelp polarity dataset and a new literature dataset that we propose. Across model variants, our metrics correlate well with human judgments, at both the sentence-level and system-level.

2 Related Work

Textual Transfer Evaluation Recent work has included human evaluation of the three categories (post-transfer style accuracy, semantic preservation, fluency), but does not propose automatic evaluation metrics for all three (Li et al., 2018; Prabhumoye et al., 2018; Chen et al., 2018; Zhang et al., 2018). There have been recent proposals for supervised evaluation metrics (Li et al., 2018), but these require annotation and are therefore unavailable for new textual transfer tasks. There is a great deal of recent work in textual transfer (Yang et al., 2018b; Santos et al., 2018; Zhang et al., 2018; Logeswaran et al., 2018; Nikolov and Hahnloser, 2018), but all either lack certain categories of unsupervised metric or lack human validation of them, which we contribute. Moreover, the textual transfer community lacks discussion of early stopping criteria and methods of holistic model comparison. We propose a one-number summary for transfer quality, which can be used to select and compare models.

In contemporaneous work, Mir et al. (2019) similarly proposed three types of metrics for style transfer tasks. There are two main differences compared to our work: (1) They use a style-keyword masking procedure before evaluating semantic similarity, which works on the Yelp dataset (the only dataset Mir et al. (2019) test on) but does not work on our Literature dataset or similarly complicated tasks, because the masking procedure goes against preserving content-specific non-style-related words. (2) They do not provide a way of aggregating three metrics for the purpose of model selection and overall comparison. We address these two problems, and we also propose metrics that are simple in addition to being effective, which is beneficial for ease of use and widespread adoption.

Textual Transfer Models In terms of generating the transferred sentences, to address the lack of parallel data, Hu et al. (2017) used variational autoencoders to generate content representations devoid of style, which can be converted to sentences with a specific style. Ficler and Goldberg (2017) used conditional language models to generate sentences where the desired content and style are conditioning contexts. Li et al. (2018) used a feature-based approach that deletes characteristic words from the original sentence, retrieves similar sentences in the target corpus, and generates based on the original sentence and the characteristic words from the retrieved sentences. Xu et al. (2018) integrated reinforcement learning into the textual transfer problem. Another way to address the lack of parallel data is to use learning frameworks based on adversarial objectives (Goodfellow et al., 2014); several have done so for textual transfer (Yu et al., 2017; Li et al., 2017; Yang et al., 2018a; Shen et al., 2017; Fu et al., 2018). Recent work uses target-domain language models as discriminators to provide more stable feedback in learning (Yang et al., 2018b).

To preserve semantics more explicitly, Fu et al. (2018) use a multi-decoder model to learn content representations that do not reflect styles. Shetty et al. (2017) use a cycle constraint that penalizes L_1 distance between input and round-trip transfer reconstruction. Our cycle consistency loss is inspired by Shetty et al. (2017), together with the idea of back translation in unsupervised neural machine translation (Artetxe et al., 2017; Lample et al., 2017), and the idea of cycle constraints in image generation by Zhu et al. (2017).

3 Evaluation

3.1 Issues with Most Existing Methods

Prior work in automatic evaluation of textual transfer has focused on post-transfer classification accuracy ("Acc"), computed by using a pretrained classifier to measure classification accuracy of transferred texts (Hu et al., 2017; Shen et al., 2017). However, there is a problem with

#ep	Acc	Sim	Sentence
original input			the host that walked us to the table and left without a word .
0.5	0.87	0.65	the food is the best and the food is the .
3.3	0.72	0.75	the owner that went to to the table and made a smile .
7.5	0.58	0.81	the host that walked through to the table and are quite perfect !

Table 1: Examples showing why Acc is insufficient. The original sentence has negative sentiment, and the goal is to transfer to positive. #ep is number of epochs trained when generating the sentence and Sim (described below) is the semantic similarity to the original sentence. High Acc is associated with low Sim.

relying solely on this metric. Table 1 shows examples of transferred sentences at several points in training the model of Shen et al. (2017). Acc is highest very early in training and decreases over time as the outputs become a stronger semantic match to the input, a trend we show in more detail in Section 6. Thus transfer quality is inversely proportional to semantic similarity to the input sentence, meaning that these metrics are complementary and difficult to optimize simultaneously.

We also identify a third category of metric, namely fluency of the transferred sentence, and similarly find it to be complementary to the first two. These three metrics can be used to evaluate textual transfer systems and to do hyperparameter tuning and early stopping. In our experiments, we found that training typically converges to a point that gives poor Acc. Intermediate results are much better under a combination of all three unsupervised metrics. Stopping criteria are rarely discussed in prior work on textual transfer.

3.2 Unsupervised Evaluation Metrics

We now describe our proposals. We validate the metrics with human judgments in Section 6.3.

Post-transfer classification accuracy ("Acc"): This metric was mentioned above. We use a CNN (Kim, 2014) trained to classify a sentence as being from $\mathbf{X}_0$ or $\mathbf{X}_1$ (two corpora corresponding to different styles or attributes). Then Acc is the percentage of transferred sentences that are classified as belonging to the transferred class.

Semantic Similarity ("Sim"): We compute semantic similarity between the input and transferred sentences. We embed sentences by averaging their word embeddings weighted by idf scores, where $\text{idf}(q) = \log(|C| \cdot |\{s \in C : q \in s\}|^{-1})$ (q is a word, s is a sentence, $C = \mathbf{X}_0 \cup \mathbf{X}_1$). We use 300-dimensional GloVe word embeddings (Pennington et al., 2014). Then, Sim is the average of the cosine similarities over all original/transferred sentence pairs. Though this metric is quite simple, we show empirically that it is effective in capturing semantic similarity. Simplicity in evaluation metrics is beneficial for computational efficiency and widespread adoption. The quality of transfer evaluations will be significantly boosted with even such a simple metric. We also experimented with METEOR (Denkowski and Lavie, 2014). However, given that we found it to be strongly correlated with Sim (shown in supplemental materials), we adopt Sim due to its computational efficiency and simplicity.

Different textual transfer tasks may require different degrees of semantic preservation. Our summary metric, described in Section 3.3, can be tailored by practitioners for various datasets and tasks which may require more or less weight on semantic preservation.

Fluency ("PP"): Transferred sentences can exhibit high Acc and Sim while still being ungrammatical. So we add a third unsupervised metric to target fluency. We compute perplexity ("PP") of the transferred corpus, using a language model pretrained on the concatenation of $\mathbf{X}_0$ and $\mathbf{X}_1$. We note that perplexity is distinct from fluency. However, certain measures based on perplexity have been shown to correlate with sentence-level human fluency judgments (Gamon et al., 2005; Kann et al., 2018). Furthermore, as discussed in Section 3.3, we punish abnormally small perplexities, as transferred texts with such perplexities typically consist entirely of words and phrases that do not result in meaningful sentences. Our summary metric, described in Section 3.3, can be tailored by practitioners for various datasets and tasks which may require more or less weight on semantic preservation.

3.3 Summarizing Metrics into One Score

It is often useful to summarize multiple metrics into one number, for ease of tuning and model selection. To do so, we propose an adjusted geometric mean (GM) of a generated sentence q:

$$\text{GM}_{\mathbf{t}}(q) = \big([100 \cdot \text{Acc} - t_1]_+ \cdot [100 \cdot \text{Sim} - t_2]_+ \cdot \min\{[t_3 - \text{PP}]_+, [\text{PP} - t_4]_+\}\big)^{\frac{1}{3}} \quad (1)$$

where $\mathbf{t} = (t_i)_{i\in[4]}$, and $[\cdot]_+ = \max(\cdot, 0)$. Note that as discussed above, we punish abnormally small perplexities by setting t_4.

When choosing models, different practitioners may prefer different trade-offs of Acc, Sim, and PP. As one example, we provide a set of parameters based on *our* experiments: $\mathbf{t} = (63, 71, 97, -37)$. We sampled 300 pairs of transferred sentences from a range of models from our two different tasks (Yelp and literature) and asked annotators which of the two sentences is better. We denote a pair of sentences by (y^+, y^-) where y^+ is preferred. We train the parameters $\mathbf{t}$ using the following loss:

$$L_{\text{GM}}(\mathbf{t}) = \max(0, -\text{GM}_{\mathbf{t}}(y^+) + \text{GM}_{\mathbf{t}}(y^-) + 1)$$

In future work, a richer function $f(\text{Acc}, \text{Sim}, \text{PP})$ could be learned from additional annotated data, and more diverse textual transfer tasks can be integrated into the parameter training.

4 Textual Transfer Models

The textual transfer systems introduced below are designed to target the metrics. These system variants are also used for metric evaluation. Note that each variant of the textual transfer system uses different components described below.

Our model is based on Shen et al. (2017). We define $\mathbf{y} \in \mathbb{R}^{200}$ and $\mathbf{z} \in \mathbb{R}^{500}$ to be latent style and content variables, respectively. $\mathbf{X}_0$ and $\mathbf{X}_1$ are two corpora containing sentences $\mathbf{x}_0^{(i)}$ and $\mathbf{x}_1^{(i)}$ respectively, where the word embeddings are in $\mathbb{R}^{100}$. We transfer using an encoder-decoder framework. The encoder $E : \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$ (where $\mathcal{X}, \mathcal{Y}, \mathcal{Z}$ are sentence domain, style space, and content space, respectively) is defined using an RNN with gated recurrent unit (GRU; Chung et al., 2014) cells. The decoder/generator $G : \mathcal{Y} \times \mathcal{Z} \to \mathcal{X}$ is defined also using a GRU RNN. We use $\widetilde{\mathbf{x}}$ to denote the style-transferred version of $\mathbf{x}$. We want $\widetilde{\mathbf{x}}_t^{(i)} = G(\mathbf{y}_{1-t}, E(\mathbf{x}_t^{(i)}, \mathbf{y}_t))$ for $t \in \{0, 1\}$.

4.1 Reconstruction and Adversarial Losses

Shen et al. (2017) used two families of losses for training: reconstruction and adversarial losses. The reconstruction loss solely helps the encoder and decoder work well at encoding and generating natural language, without any attempt at transfer:

$$L_{rec}(\theta_E, \theta_G) = \sum_{t=0}^{1} \mathbb{E}_{\mathbf{x}_t}\big[-\log p_G(\mathbf{x}_t \mid \mathbf{y}_t, E(\mathbf{x}_t, \mathbf{y}_t))\big] \quad (2)$$

The loss seeks to ensure that when a sentence $\mathbf{x}_t$ is encoded to its content vector and then decoded to generate a sentence, the generated sentence should match $\mathbf{x}_t$. For their adversarial loss, Shen et al. (2017) used a pair of discriminators: D_0 tries to distinguish between $\mathbf{x}_0$ and $\widetilde{\mathbf{x}}_1$, and D_1 between $\mathbf{x}_1$ and $\widetilde{\mathbf{x}}_0$. In particular, decoder G's hidden states are aligned instead of output words.

$$L_{adv_t}(\theta_E, \theta_G, \theta_{D_t}) = -\tfrac{1}{k}\sum_{i=1}^{k} \log D_t(\mathbf{h}_t^{(i)}) - \tfrac{1}{k}\sum_{i=1}^{k} \log(1 - D_t(\widetilde{\mathbf{h}}_{1-t}^{(i)})) \quad (3)$$

where k is the size of a mini-batch. D_t outputs the probability that its input is from style t where the classifiers are based on the convolutional neural network from Kim (2014). The CNNs use filter n-gram sizes of 3, 4, and 5, with 128 filters each. We obtain hidden states $\mathbf{h}$ by unfolding G from the initial state $(\mathbf{y}_t, \mathbf{z}_t^{(i)})$ and feeding in $\mathbf{x}_t^{(i)}$. We obtain hidden states $\widetilde{\mathbf{h}}$ by unfolding G from $(\mathbf{y}_{1-t}, \mathbf{z}_t^{(i)})$ and feeding in the previous output probability distributions.

4.2 Cyclic Consistency Loss

We use a "cyclic consistency" loss (Zhu et al., 2017) to encourage already-transferred sentences to be able to be recovered by transferring back again. This loss is similar to L_{rec} except we now transfer style twice in the loss. Recall that we seek to transfer $\mathbf{x}_t$ to $\widetilde{\mathbf{x}}_t$. After successful transfer, we expect $\widetilde{\mathbf{x}}_t$ to have style $\mathbf{y}_{1-t}$, and $\widetilde{\widetilde{\mathbf{x}}}_t$ (transferred back from $\widetilde{\mathbf{x}}_t$) to have style $\mathbf{y}_t$. We want $\widetilde{\widetilde{\mathbf{x}}}_t$ to be very close to the original untransferred $\mathbf{x}_t$. The loss is defined as

$$L_{cyc}(\theta_E, \theta_G) = \sum_{t=0}^{1} \mathbb{E}_{\mathbf{x}_t}\big[-\log p_G(\mathbf{x}_t \,|\, \mathbf{y}_t, \widetilde{\mathbf{z}}_t)\big] \quad (4)$$

where $\widetilde{\mathbf{z}}_t = E(G(\mathbf{y}_{1-t}, E(\mathbf{x}_t, \mathbf{y}_t)), \mathbf{y}_{1-t})$ or, more concisely, $\widetilde{\mathbf{z}}_t = E(\widetilde{\mathbf{x}}_t, \mathbf{y}_{1-t})$.

To use this loss, the first step is to transfer sentences $\mathbf{x}_t$ from style t to $1-t$ to get $\widetilde{\mathbf{x}}_t$. The second step is to transfer $\widetilde{\mathbf{x}}_t$ of style $1-t$ back to t so that we can compute the loss of the words in $\mathbf{x}_t$ using probability distributions computed by the decoder. Backpropagation on the embedding, encoder, and decoder parameters will only be based on the second step, because the first step involves argmax operations which prevent backpropagation. Still, we find that the cyclic loss greatly improves semantic preservation during transfer.

4.3 Paraphrase Loss

While L_{rec} provides the model with one way to preserve style (i.e., simply reproduce the input), the model does not see any examples of style-preserving paraphrases. To address this, we add a paraphrase loss very similar to losses used in neural machine translation. We define the loss on a sentential paraphrase pair $\langle \mathbf{u}, \mathbf{v} \rangle$ and assume that $\mathbf{u}$ and $\mathbf{v}$ have the same style and content. The loss is the sum of token-level log losses for generating each word in $\mathbf{v}$ conditioned on the encoding of $\mathbf{u}$:

$$L_{para}(\theta_E, \theta_G) = \sum_{t=0}^{1} \mathbb{E}_{\langle \mathbf{u},\mathbf{v} \rangle} \left[-\log p_G(\mathbf{v} \mid \mathbf{y}_t, E(\mathbf{u}, \mathbf{y}_t)) \right] \quad (5)$$

For paraphrase pairs, we use the ParaNMT-50M dataset (Wieting and Gimpel, 2018).[1]

4.4 Language Modeling Loss

We attempt to improve fluency (our third metric) and assist transfer with a loss based on matching a pretrained language model for the target style. The loss is the cross entropy (CE) between the probability distribution from this language model and the distribution from the decoder:

$$L_{lang}(\theta_E, \theta_G) = \sum_{t=0}^{1} \mathbb{E}_{\mathbf{x}_t} \left[\sum_i \mathrm{CE}(\mathbf{l}_{t,i}, \mathbf{g}_{t,i}) \right] \quad (6)$$

where $\mathbf{l}_{t,i}$ and $\mathbf{g}_{t,i}$ are distributions over the vocabulary defined as follows:

$$\mathbf{l}_{t,i} = p_{LM_{1-t}}(\cdot \mid \widetilde{\mathbf{x}}_{t_{1:(i-1)}})$$
$$\mathbf{g}_{t,i} = p_G(\cdot \mid \widetilde{\mathbf{x}}_{t_{1:(i-1)}}, \mathbf{y}_{1-t}, E(\mathbf{x}_t, \mathbf{y}_t))$$

where $\cdot$ stands for all words in the vocabulary built from the corpora. When transferring from style t to $1-t$, $\mathbf{l}_{t,i}$ is the distribution under the language model $p_{LM_{1-t}}$ pretrained on sentences from style $1-t$ and $\mathbf{g}_{t,i}$ is the distribution under the decoder G. The two distributions $\mathbf{l}_{t,i}$ and $\mathbf{g}_{t,i}$ are over words at position i given the $i-1$ words already predicted by the decoder. The two style-specific language models are pretrained on the corpora corresponding to the two styles. They are GRU RNNs with a dropout probability of 0.5, and they are kept fixed during the training of the transfer network.

4.5 Multiple Discriminators

Note that each of the textual transfer system variants uses different losses or components described in this section. To create more variants, we add a second pair of discriminators, D'_0 and D'_1, to the adversarial loss to address the possible mode collapse problem (Nguyen et al., 2017). In particular, we use CNNs with n-gram filter sizes of 3, 4, and 5 for D_0 and D_1, and we use CNNs with n-gram sizes of 1, 2, and 3 for D'_0 and D'_1. Also, for D'_0 and D'_1, we use the Wasserstein GAN (WGAN) framework (Arjovsky et al., 2017). The adversarial loss takes the following form:

$$L_{adv'_t}(\theta_E, \theta_G, \theta_{D'_t}) = \tfrac{1}{k} \sum_{i=1}^{k} \big[D'_t(\widetilde{\mathbf{h}}_t^{(i)}) - D'_t(\mathbf{h}_t^{(i)}) + \xi(\|\nabla_{\mathring{\mathbf{h}}_t^{(i)}} D'_t(\mathring{\mathbf{h}}_t^{(i)})\|_2 - 1)^2 \big] \quad (7)$$

where $\mathring{\mathbf{h}}_t^{(i)} = \epsilon_i \mathbf{h}_t^{(i)} + (1-\epsilon_i)\widetilde{\mathbf{h}}_t^{(i)}$ where $\epsilon_i \sim \mathrm{Uniform}([0,1])$ is sampled for each training instance. The adversarial loss is based on Arjovsky et al. (2017),[2] with the exception that we use the hidden states of the decoder instead of word distributions as inputs to D'_t, similar to Eq. (3).

We choose WGAN in the hope that its differentiability properties can help avoid vanishing gradient and mode collapse problems. We expect the generator to receive helpful gradients even if the discriminators perform well. This approach leads to much better outputs, as shown below.

4.6 Summary

We iteratively update (1) θ_{D_0}, θ_{D_1}, $\theta_{D'_0}$, and $\theta_{D'_1}$ by gradient descent on L_{adv_0}, L_{adv_1}, $L_{adv'_0}$, and $L_{adv'_1}$, respectively, and (2) θ_E, θ_G by gradient descent on $L_{total} = \lambda_1 L_{rec} + \lambda_2 L_{para} + \lambda_3 L_{cyc} + \lambda_4 L_{lang} - \lambda_5(L_{adv_0} + L_{adv_1}) - \lambda_6(L_{adv'_0} + L_{adv'_1})$. Depending on which model is being trained (see Table 2), the λ_i's for the unused losses will be zero. More details are shown in Section 5. The appendix shows the full algorithm.

5 Experimental Setup

5.1 Datasets

Yelp sentiment. We use the same Yelp dataset as Shen et al. (2017), which uses corpora of positive and negative Yelp reviews. The goal of the transfer task is to generate rewritten sentences with similar content but inverted sentiment. We use the same train/development/test split as Shen et al. (2017). The dataset has 268K, 38K, 76K positive training, development, and test sentences, respectively, and 179K/25K/51K negative sentences. Like Shen

[1] We first filter out sentence pairs where one sentence is the substring of another, and then randomly select 90K pairs.

[2] We use a default value of $\xi = 10$.

et al. (2017), we only use sentences with 15 or fewer words.

Literature. We consider two corpora of literature. The first corpus contains works of Charles Dickens collected from Project Gutenberg. The second corpus is comprised of modern literature from the Toronto Books Corpus (Zhu et al., 2015). Sentences longer than 25 words are removed. Unlike the Yelp dataset, the two corpora have very different vocabularies. This dataset poses challenges for the textual transfer task, and it provides diverse data for assessing quality of our evaluation system. Given the different and sizable vocabulary, we preprocess by using the named entity recognizer in Stanford CoreNLP (Manning et al., 2014) to replace names and locations with -PERSON- and -LOCATION- tags, respectively. We also use byte-pair encoding (BPE), commonly used in generation tasks (Sennrich et al., 2016b). We only use sentences with lengths between 6 and 25. The resulting dataset has 156K, 5K, 5K Dickens training, development, and testing sentences, respectively, and 165K/5K/5K modern literature sentences.

5.2 Hyperparameter Settings

Section 4.6 requires setting the λ weights for each component. Depending on which model is being trained (see Table 2), the λ_i's for the unused losses will be zero. Otherwise, we set $\lambda_1 = 1$, $\lambda_2 = 0.2$, $\lambda_3 = 5$, $\lambda_4 = 10^{-3}$, $\lambda_5 = 1$, $\lambda_6 = 2^{-ep}$ where ep is the number of epochs. For optimization we use Adam (Kingma and Ba, 2014) with a learning rate of 10^{-4}. We implement our models using TensorFlow (et al., 2015).[3] Code is available via the first author's webpage `yzpang.me`.

5.3 Pretrained Evaluation Models

For the pretrained classifiers, the accuracies on the Yelp and Literature development sets are 0.974 and 0.933, respectively. For language models, the perplexities on the Yelp and Literature development sets are 27.4 and 40.8, respectively.

6 Results and Analysis

6.1 Analyzing Metric Relationships

Table 2 shows results for the Yelp dataset and Figure 1 plots learning trajectories of those models.

[3]Our implementation is based on code from Shen et al. (2017).

	Acc	Sim	PP	GM
M0: Shen et al. (2017)	0.818	0.719	37.3	10.0
M1: M0+*para*	0.819	0.734	26.3	14.2
M2: M0+*cyc*	0.813	0.770	36.4	18.8
M3: M0+*cyc*+*lang*	0.807	0.796	28.4	21.5
M4: M0+*cyc*+*para*	0.798	0.783	39.7	19.2
M5: M0+*cyc*+*para*+*lang*	0.804	0.785	27.1	20.3
M6: M0+*cyc*+*2d*	0.805	**0.817**	43.3	21.6
M7: M6+*para*+*lang*	0.818	0.805	**29.0**	**22.8**

Table 2: Yelp results with various systems and automatic metrics at a nearly-fixed Acc, with best scores in boldface. We use M0 to denote Shen et al. (2017).

	Acc	Sim	PP	GM
M0: Shen et al. (2017)	0.694	0.728	**22.3**	8.81
M1: M0+*para*	0.702	0.747	23.6	11.7
M2: M0+*cyc*	0.692	0.781	49.9	**12.8**
M3: M0+*cyc*+*lang*	0.698	0.754	39.2	12.0
M4: M0+*cyc*+*para*	0.702	0.757	33.9	**12.8**
M5: M0+*cyc*+*para*+*lang*	0.688	0.753	28.6	11.8
M6: M0+*cyc*+*2d*	0.704	**0.794**	63.2	**12.8**
M7: M6+*para*+*lang*	0.706	0.768	49.0	**12.8**

Table 3: Literature results with various systems and automatic metrics at a nearly-fixed Acc, with best scores in boldface. We use M0 to denote Shen et al. (2017).

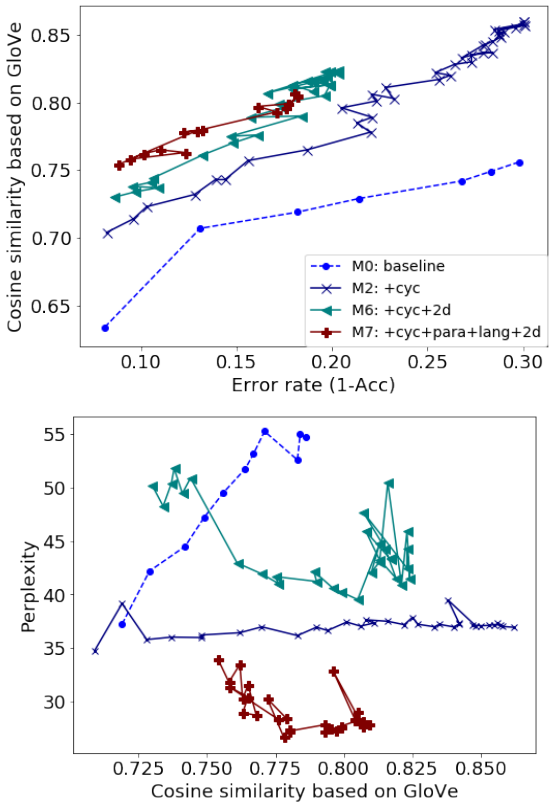

Figure 1: Learning trajectories with models from Table 2. Metrics are computed on the dev sets. Figures for Literature (with similar trends) are in supplementary.

Table 3 shows results for the Literature dataset. Models for the Literature dataset show similar

Dataset	Models		Transfer quality			Semantic preservation				Fluency			
	A	B	A>B	B>A	Tie	A>B	B>A	Tie	Δ_{Sim}	A>B	B>A	Tie	Δ_{PP}
Yelp	M0	M2	9.0	6.0	85.1	1.5	**25.4**	73.1	-0.05	10.4	**23.9**	65.7	0.9
	M0	M7	9.6	14.7	75.8	2.5	**54.5**	42.9	-0.09	4.6	**39.4**	56.1	8.3
	M6	M7	13.7	11.6	74.7	16.0	16.7	67.4	0.01	10.3	20.0	69.7	14.3
	M2	M7	5.8	9.3	84.9	8.1	**25.6**	66.3	-0.04	14.0	**26.7**	59.3	7.4
Literature	M2	M6	4.2	6.7	89.2	16.7	20.8	62.5	0.01	**40.8**	13.3	45.8	-13.3
	M6	M7	15.8	13.3	70.8	**25.0**	9.2	65.8	0.03	14.2	20.8	65.0	14.2

Table 4: Manual evaluation results (%) using models from Table 2 (i.e., with roughly fixed Acc). > means "better than". $\Delta_{\text{Sim}} = \text{Sim}(A) - \text{Sim}(B)$, and $\Delta_{\text{PP}} = \text{PP}(A) - \text{PP}(B)$ (note that lower PP generally means better fluency). Each row uses at least 120 sentence pairs. A cell is bold if it represents a model win of at least 10%.

trends. The figures show trajectories of statistics on corpora transferred/generated from the dev set during learning. Each two consecutive markers deviate by half an epoch of training. Lower-left markers generally precede upper-right ones. In Figure 1(a), the plots of Sim by error rate (1 − Acc) exhibit positive slopes, meaning that error rate is positively correlated with Sim. Curves to the upper-left corner represent better trade-off between error rate and Sim. In the plots of PP by Sim in Figure 1(b), the M0 curve exhibits large positive slope but the curves for other models do not, which indicates that M0 sacrifices PP for Sim. Other models maintain consistent PP as Sim increases during training.

6.2 System-Level Validation

Annotators were shown the untransferred sentence, as well as sentences produced by two models (which we refer to as A and B). They were asked to judge which better reflects the target style (A, B, or tie), which has better semantic preservation of the original (A, B, or tie), and which is more fluent (A, B, or tie). Results are shown in Table 4.

Overall, the results show the same trends as our automatic metrics. For example, on Yelp, large differences in human judgments of semantic preservation (M2>M0, M7>M0, M7>M2) also show the largest differences in Sim, while M6 and M7 have very similar human judgments and very similar Sim scores.

6.3 Sentence-Level Validation of Metrics

We describe a human sentence-level validation of our metrics in Table 5.

To validate Acc, human annotators were asked to judge the style of 100 transferred sentences (sampled equally from M0, M2, M6, M7). Note that it is a binary choice question (style 0 or style 1 without "tie" option) so that human annotators had to make a choice. We then compute the percentage of machine and human judgments that match.

Metric	Method of validation	Yelp	Lit.
Acc	% of machine and human judgments that match	94	84
Sim	Spearman's ρ b/w Sim and human ratings of semantic preservation	0.79	0.75
PP	Spearman's ρ b/w negative PP and human ratings of fluency	0.81	0.67

Table 5: Human sentence-level validation of metrics; 100 examples for each dataset for validating Acc; 150 each for Sim and PP; see text for validation of GM.

We validate Sim and PP by computing sentence-level Spearman's ρ between the metric and human judgments (an integer score from 1 to 4) on 150 generated sentences (sampled equally from M0, M2, M6, M7). We presented pairs of original sentences and transferred sentences to human annotators. They were asked to rate the level of semantic similarity (and similarly for fluency) where 1 means "extremely bad", 2 means "bad/ok/needs improvement", 3 means "good", and 4 means "very good." They were also given 5 examples for each rating (i.e., a total of 20 for four levels) before annotating. From Table 5, all validations show strong correlations on the Yelp dataset and reasonable correlations on Literature.

We validate GM by obtaining human pairwise preferences (without the "tie" option) of overall transfer quality and measuring the fraction of pairs in which the GM score agrees with the human preference. Out of 300 pairs (150 from each dataset), 258 (86%) match.

The transferred sentences used in the evaluation are sampled from the development sets produced by models M0, M2, M6, and M7, at the accuracy levels used in Table 2. In the data preparation for

the manual annotation, there is sufficient randomization regarding model and textual transfer direction.

6.4 Comparing Losses

Cyclic Consistency Loss. We compare the trajectories of the baseline model (M0) and the *+cyc* model (M2). Table 2 and Figure 1 show that under similar Acc, M2 has much better semantic similarity for both Yelp and Literature. In fact, cyclic consistency loss proves to be the strongest driver of semantic preservation across all of our model configurations. The other losses do not constrain the semantic relationship across style transfer, so we include the cyclic loss in M3 to M7.

Paraphrase Loss. Table 2 shows that the model with paraphrase loss (M1) slightly improves Sim over M0 on both datasets under similar Acc. For Yelp, M1 has better Acc and PP than M0 at comparable semantic similarity. So, when used alone, the paraphrase loss helps. However, when combined with other losses (e.g., compare M2 to M4), its benefits are mixed. For Yelp, M4 is slightly better in preserving semantics and producing fluent output, but for Literature, M4 is slightly worse. A challenge in introducing an additional paraphrase dataset is that its notions of similarity may clash with those of content preservation in the transfer task. For Yelp, both corpora share a great deal of semantic content, but Literature shows systematic semantic differences even after preprocessing.

Language Modeling Loss. When comparing between M2 and M3, between M4 and M5, and between M6 and M7, we find that the addition of the language modeling loss reduces PP, sometimes at a slight cost of semantic preservation.

6.5 Results based on Supervised Evaluation

If we want to compare the models using one single number, GM is our unsupervised approach. We can also compute BLEU scores between our generated outputs and human-written gold standard outputs using the 1000 Yelp references from Li et al. (2018). For BLEU scores reported for the methods of Li et al. (2018), we use the values reported by Yang et al. (2018b). We use the same BLEU implementation as used by Yang et al. (2018b), i.e., `multi-bleu.perl`. We compare three models selected during training from each of our M6 and M7 settings. We also report post-transfer accuracies reported by prior work, as well as our own computed Acc scores for M0, M6, M7, and the untransferred sentences. Though the classifiers differ across models, their accuracy tends to be very high (> 0.97), making it possible to make rough comparisons of Acc across them.

Model	BLEU	Acc*
Fu et al. (2018)		
Multi-decoder	7.6	0.792
Style embed.	15.4	0.095
Li et al. (2018)		
Template	18.0	0.867
Delete/Retrieve	12.6	0.909
Yang et al. (2018b)		
LM	13.4	0.854
LM + classifier	**22.3**	0.900
Untransferred	**31.4**	0.024

M.	BLEU	Acc
M0	4.9	0.818
M6	22.3	0.804
M6	**22.5**	0.843
M6	16.3	0.897
M7	17.0	0.814
M7	16.3	0.839
M7	12.9	0.901

Table 6: Results on Yelp sentiment transfer, where BLEU is between 1000 transferred sentences and human references, and Acc is restricted to the same 1000 sentences. Our best models (right table) achieve higher BLEU than prior work at similar levels of Acc, but untransferred sentences achieve the highest BLEU. Acc*: the definition of Acc varies by row because of different classifiers in use. Other results from Li et al. (2018) are not included as they are worse.

BLEU scores and post-transfer accuracies are shown in Table 6. The most striking result is that *untransferred* sentences have the highest BLEU score by a large margin, suggesting that prior work for this task has not yet eclipsed the trivial baseline of returning the input sentence. However, at similar levels of Acc, our models have higher BLEU scores than prior work. We additionally find that supervised BLEU shows a trade-off with Acc: for a single model type, higher Acc generally corresponds to lower BLEU.

7 Conclusion

We proposed three kinds of metrics for non-parallel textual transfer, studied their relationships, and developed learning criteria to address them. We emphasize that all three metrics are needed to make meaningful comparisons among models. We expect our components to be applicable to a broad range of generation tasks.

Acknowledgments

We thank Karl Stratos and Zewei Chu for helpful discussions, the annotators for performing manual evaluations, and the anonymous reviewers for useful comments. We also thank Google for a faculty research award to K. Gimpel that partially supported this research.

References

Martín Abadi et al. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein GAN. *arXiv preprint arXiv:1701.07875*.

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

Liqun Chen, Shuyang Dai, Chenyang Tao, Haichao Zhang, Zhe Gan, Dinghan Shen, Yizhe Zhang, Guoyin Wang, Ruiyi Zhang, and Lawrence Carin. 2018. Adversarial text generation via feature-mover's distance. In *Advances in Neural Information Processing Systems*, pages 4671–4682.

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

Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the EACL 2014 Workshop on Statistical Machine Translation*.

Jessica Ficler and Yoav Goldberg. 2017. Controlling linguistic style aspects in neural language generation. In *Proceedings of the Workshop on Stylistic Variation*, pages 94–104, Copenhagen, Denmark. Association for Computational Linguistics.

Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan. 2018. Style transfer in text: exploration and evaluation. In *32nd AAAI Conference on Artificial Intelligence (AAAI-18)*.

Michael Gamon, Anthony Aue, and Martine Smets. 2005. Sentence-level MT evaluation without reference translations: Beyond language modeling. In *Proceedings of EAMT*, pages 103–111.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680.

George Heidorn. 2000. Intelligent writing assistance. *Handbook of natural language processing*, pages 181–207.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P. Xing. 2017. Toward controlled generation of text. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1587–1596.

Katharina Kann, Sascha Rothe, and Katja Filippova. 2018. Sentence-level fluency evaluation: References help, but can be spared! In *Proceedings of the 22nd Conference on Computational Natural Language Learning, CoNLL 2018, Brussels, Belgium, October 31 - November 1, 2018*, pages 313–323.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

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

Jiwei Li, Will Monroe, Tianlin Shi, Sébastien Jean, Alan Ritter, and Dan Jurafsky. 2017. Adversarial learning for neural dialogue generation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 2147–2159. Association for Computational Linguistics.

Juncen Li, Robin Jia, He He, and Percy Liang. 2018. Delete, retrieve, generate: a simple approach to sentiment and style transfer. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1865–1874. Association for Computational Linguistics.

Lajanugen Logeswaran, Honglak Lee, and Samy Bengio. 2018. Content preserving text generation with attribute controls. In *Advances in Neural Information Processing Systems*, pages 5103–5113.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland. Association for Computational Linguistics.

Remi Mir, Bjarke Felbo, Nick Obradovich, and Iyad Rahwan. 2019. Evaluating style transfer for text. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 495–504, Minneapolis, Minnesota. Association for Computational Linguistics.

Tu Nguyen, Trung Le, Hung Vu, and Dinh Phung. 2017. Dual discriminator generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2667–2677.

Nikola I Nikolov and Richard HR Hahnloser. 2018. Large-scale hierarchical alignment for author style transfer. *arXiv preprint arXiv:1810.08237*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Shrimai Prabhumoye, Yulia Tsvetkov, Ruslan Salakhutdinov, and Alan W Black. 2018. Style transfer through back-translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 866–876, Melbourne, Australia. Association for Computational Linguistics.

Sudha Rao and Joel Tetreault. 2018. Dear sir or madam, may i introduce the gyafc dataset: Corpus, benchmarks and metrics for formality style transfer. *arXiv preprint arXiv:1803.06535*.

Alan Ritter, Colin Cherry, and William B Dolan. 2011. Data-driven response generation in social media. In *Proceedings of the conference on empirical methods in natural language processing*, pages 583–593. Association for Computational Linguistics.

Cicero Nogueira dos Santos, Igor Melnyk, and Inkit Padhi. 2018. Fighting offensive language on social media with unsupervised text style transfer. *arXiv preprint arXiv:1805.07685*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Controlling politeness in neural machine translation via side constraints. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 35–40. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725. Association for Computational Linguistics.

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment. In *Advances in Neural Information Processing Systems 30*, pages 6833–6844. Curran Associates, Inc.

Rakshith Shetty, Bernt Schiele, and Mario Fritz. 2017. Author attribute anonymity by adversarial training of neural machine translation. *arXiv preprint arXiv:1711.01921*.

John Wieting and Kevin Gimpel. 2018. ParaNMT-50M: Pushing the limits of paraphrastic sentence embeddings with millions of machine translations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Jingjing Xu, Xu Sun, Qi Zeng, Xiaodong Zhang, Xuancheng Ren, Houfeng Wang, and Wenjie Li. 2018. Unpaired sentiment-to-sentiment translation: A cycled reinforcement learning approach. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 979–988, Melbourne, Australia. Association for Computational Linguistics.

Zhen Yang, Wei Chen, Feng Wang, and Bo Xu. 2018a. Improving neural machine translation with conditional sequence generative adversarial nets. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1346–1355, New Orleans, Louisiana. Association for Computational Linguistics.

Zichao Yang, Zhiting Hu, Chris Dyer, Eric P Xing, and Taylor Berg-Kirkpatrick. 2018b. Unsupervised text style transfer using language models as discriminators. *arXiv preprint arXiv:1805.11749*.

Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. Seqgan: Sequence generative adversarial nets with policy gradient. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Yi Zhang, Jingjing Xu, Pengcheng Yang, and Xu Sun. 2018. Learning sentiment memories for sentiment modification without parallel data. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1103–1108, Brussels, Belgium. Association for Computational Linguistics.

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

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 19–27.