Learning Criteria and Evaluation Metrics for Textual Transfer between Non-Parallel Corpora

Yuanzhe Pang
University of Chicago
Chicago, IL 60637
yzpang@uchicago.edu

Kevin Gimpel
Toyota Technological Institute at Chicago
Chicago, IL 60637
kgimpel@ttic.edu

Abstract

We consider the problem of automatically generating textual paraphrases with modified attributes or stylistic properties, focusing on the setting without parallel data (Hu et al., 2017; Shen et al., 2017). This setting poses challenges for learning and evaluation. We show that the metric of post-transfer classification accuracy is insufficient on its own, and propose additional metrics based on semantic content preservation and fluency. For reliable evaluation, all three metric categories must be taken into account. We contribute new loss functions and training strategies to address the new 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. Automatic and manual evaluation show large improvements over the baseline method of Shen et al. (2017). Our hope is that these losses and metrics can be general and useful tools for a range of textual transfer settings without parallel corpora.

1 Introduction

We focus on the problem of textual transfer, which we define as the ability to automatically generate textual paraphrases with modified attributes or stylistic properties. An effective textual transfer system could benefit a range of user-facing text generation applications, including dialogue, message response generation (Ritter et al., 2011; Li et al., 2016; Wen et al., 2016), writing assistance (Heidorn, 2000), user adaptation in educational applications (Campbell, 1987), and machine translation (Bahdanau et al., 2014). Textual transfer could also be used to improve NLP systems by providing new methods for data augmentation and domain adaptation.

One factor that makes textual transfer difficult is the lack of parallel corpora. We follow recent work in focusing on the setting in which we have two corpora representing different attributes or styles but do not have any parallel data between them (Hu et al., 2017; Shen et al., 2017).

Style transfer has been evaluated automatically by running a pretrained style classifier on the output and computing 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 using the model of Shen et al. (2017). We address this deficiency of post-transfer accuracy by identifying two competing goals: preserving semantic content and producing fluent output. We wish to preserve aspects of the source text that pertain to content and transfer aspects that pertain to style, while also producing fluent and coherent output.

In this paper, we contribute automatic evaluation metrics for these additional desiderata and learning criteria to target them. Our goal is to propose a useful set of metrics that are also unsupervised, i.e., they do not require human-written transferred sentences. This means that our metrics can be used directly for tuning and model selection, even on test data. For semantic preservation, we consider both Meteor (Denkowski and Lavie, 2014) and cosine similarity of sentence embeddings produced by weighted word averaging. For fluency, we use perplexity under pretrained language models. The three metric categories are complementary and help us avoid degenerate behavior in tuning and model selection. We also show that intermediate points in training typically lead to generated output that is balanced among the three metrics. We perform model selection according to all three metrics jointly. For particular
applications, practitioners can choose the appropriate combination of our metrics to achieve the desired balance among transfer, semantic preservation, and fluency.

We add learning criteria to the framework of Shen et al. (2017) to accord with our new metrics. We encourage semantic preservation by adding a “cyclic consistency” loss (to ensure that transfer is reversible) and a loss defined on a separate corpus of 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 this improves the trade-off between post-transfer accuracy and semantic preservation.

Our experiments use the Yelp sentiment dataset (Shen et al., 2017) as well as a new literature dataset that we propose. The latter uses the works of Charles Dickens as one corpus and modern literature as the other. We show that our models outperform the baseline model of Shen et al. by a large margin. Under similar post-transfer accuracy, our models generate output with much higher semantic similarity. With similar semantic similarities, our models generate output with much lower perplexity. We also use a small-scale manual evaluation to corroborate our results. While we added our components to the model of Shen et al. in this paper, there are several other methods for non-parallel textual transfer and our components are applicable to many of these methods as well.

2 Related Work

There is a growing body of research in textual transfer tasks. They include diverse applications. Reddy and Knight (2016) used heuristic manipulation to transfer gender properties. Sennrich et al. (2016) used neural machine translation to transfer politeness. Hu et al. (2017) transfer sentiment and grammatical tense, Yamagishi et al. (2016) transfer between passive and active voices, Shen et al. (2017) transfer sentiment, and Shetty et al. (2017) transfer authorship so as to achieve anonymity.

To address the lack of parallel data, Hu et al. (2017) use variational autoencoders to generate content representations devoid of style, which can be converted to sentences with a specific style. Ficler and Goldberg (2017) use conditioned language models to generate sentences where the desired content and style are conditioning contexts. Another way to address the lack of parallel data is to use richer learning frameworks based on adversarial objectives (Goodfellow et al., 2014); several have used adversarial training to generate text (Yu et al., 2016; Li et al., 2017; Yang et al., 2017; Shen et al., 2017). Recent work has used target-domain language model as the discriminator to provide more stable feedback in learning. (Yang et al., 2018).

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).

There has also been a flurry of contemporary work in textual style transfer (Fu et al., 2018; Rao and Tetreault, 2018; Xu et al., 2018; Li et al., 2018). Our goal is to contribute general components—loss functions and evaluation criteria—that we hope can complement these and other studies in non-parallel text transfer.

3 Formulation and Model

Our formulation is based on Shen et al. (2017). We define \(y \in \mathcal{Y}\) and \(z \in \mathcal{Z}\) to be latent style\(^1\) and content variables, respectively, where \(\mathcal{Y} = \mathbb{R}^{200}\) and \(\mathcal{Z} = \mathbb{R}^{500}\). \(x_0\) and \(x_1\) are two corpora containing sentences \(x_0(i)\) and \(x_1(i)\) respectively, where each word is represented by a 100-dimensional embedding. That is, \(x_0(i), x_1(i) \in \mathcal{X} = \mathbb{R}^{1 \times 100}\) where \(l\) is the maximum length of a sentence.

We transfer using an encoder-decoder framework. The encoder is defined as a function \(E : \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}\) using a recurrent neural network (RNN) with gated recurrent unit (GRU; Chung et al., 2014) cells. Dropout is applied to both the input and output of the GRU cells with a dropout rate of 0.5. The decoder\(^2\) (i.e., the generator) is

\[\begin{align*}
    w_{0} &= \sigma (\mathcal{U} z + \mathcal{V} y) \\
    \text{encoder} &= \text{GRU}(x, w_{0}) \\
    \text{decoder} &= \text{GRU}(z, w_{0}) \\
    \hat{x} &= \text{softmax}(\mathcal{W} (\text{GRU}(z, w_{0})))
\end{align*}\]

\(^{1}\)We adopt an operational definition of “style” in this paper by defining it merely in terms of the difference between two corpora provided by the practitioner. If the two corpora are positive and negative Yelp reviews (Shen et al., 2017), then style is roughly equivalent to sentiment.

\(^{2}\)We use \(G\) instead of \(D\) to denote the decoder because \(D\) is reserved for the discriminator.
defined as $G : \mathcal{Y} \times Z \to \mathcal{X}$ also using a GRU RNN. We use $\tilde{x}$ to denote the style-transferred version of $x$. That is, $\tilde{x}^{(i)}_t = G(y_{1-t}, E(x^{(i)}_t, y_t))$ for $t \in \{0, 1\}$.

### 3.1 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:

$$L_{rec}(\theta_E, \theta_G) = \sum_{t=0}^{1} \mathbb{E}_{x_t}[\log p_G(x_t | y_t, E(x_t, y_t))]$$

where $p_G$ is the probability under the decoder $G$ and $\theta_E, \theta_G$ are the parameters of the encoder and decoder, respectively. The loss seeks to ensure that when a sentence $x_t$ is encoded to its content vector and then decoded to generate a sentence, the generated sentence should match $x_t$. For the adversarial loss, Shen et al. (2017) used two families of losses for training: reconstruction and adversarial losses.

When using this loss, the first step is to transfer sentences $x_t$ from style $t$ to style $1 - t$ to get $\tilde{x}_t$. The second step is to transfer $\tilde{x}_t$ of style $1 - t$ back to style $t$ so that we can compute the loss of the words in $x_t$ using the 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 prevents backpropagation. Algorithm 1 provides details.

### 4.2 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 $(u, v)$ and assume that $u$ and $v$ have the same style and content. The loss is the sum of token-level log losses for generating each word in $v$ conditioned on the encoding of $u$:

$$L_{para}(\theta_E, \theta_G) = \sum_{t=0}^{1} \mathbb{E}_{(u, v)}[\log p_G(v | y_t, E(u, y_t))]$$

For paraphrase pairs, we use the automatically-backtranslated sentence pairs in the ParaNMT-50M dataset (Wieting and Gimpel, 2018).³

### 4.3 Language Modeling Loss

Our third metric discussed in Section 1 is fluency. We attempt to improve fluency (and assist transfer) with a loss based on matching a pretrained language model for the target style. The loss is the

³We first filter out sentence pairs where one sentence is the substring of another, and then randomly select 90K pairs.
Update end

eling losses (one for each textual transfer direction) from the corpora. Each of the two language models (one for each style) distribution, i.e., the words in the vocabulary we build where $g$ to style $t$ is the probability distribution of the next word based on the decoder:

$$L_{\text{lang}}(\theta_E, \theta_G) = \sum_{t=0}^{1} \mathbb{E}_{x_t} \left[ \sum_{i} \text{CE}(l_{t,i}, g_{t,i}) \right]$$

(5)

where $l_{t,i}$ and $g_{t,i}$ are distributions over the vocabulary defined as follows:

$$l_{t,i} = p_{LM_{1-t}}( \cdot | \tilde{x}_{t,(i-1)})$$

$$g_{t,i} = p_{G}( \cdot | \tilde{x}_{t,(i-1)}, y_{1-t}, E(x_t, y_t))$$

where $\cdot$ stands for all possible values in the distribution, i.e., the words in the vocabulary we build from the corpora. Each of the two language modeling losses (one for each textual transfer direction) is comprised of cross-entropies between $l_{t,i}$ and $g_{t,i}$, summing over all sentences $x_t$ and all word positions $i$. In the context of transferring from style $t$ to style $1-t$, $l_{t,i}$ is the probability distribution of the next word based on a pretrained language model $p_{LM_{1-t}}$ for target style $1-t$, given already predicted words; and $g_{t,i}$ is the probability distribution of the next word based on the decoder $G$ in our textual transfer network, also given already predicted words.

The two language models (one for each style) are pretrain 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.

### 4.4 Multiple Discriminators

In addition to the losses proposed above, we add a second set 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 will take the following form.

$$L_{\text{adv}}'(\theta_E, \theta_G, \theta_{D_0}, \theta_{D_1}) = \frac{1}{k} \sum_{i=1}^{k} \left[ D_0'(\tilde{h}_t^{(i)}) - D_1'(\tilde{h}_t^{(i)}) + \xi (\| \nabla_{\tilde{h}_t^{(i)}} D_0'(\tilde{h}_t^{(i)}) \|_2 - 1)^2 \right]$$

(6)

where $\tilde{h}_t^{(i)} = \epsilon_i h_t^{(i)} + (1 - \epsilon_i) \tilde{h}_t^{(i)}$. The coefficient $\epsilon_i \sim \text{Uniform}[0, 1]$ is sampled for each training instance. For $\xi$, we use a default value of 10 provided by Arjovsky et al. (2017). The adversarial loss is based on Arjovsky et al. (2017), with the exception that we use the hidden states of the generator instead of word distributions as inputs to $D_t'$, similar to Eq. (2). Algorithm 1 provides details about how hidden states $h_t$ and transferred hidden states $\tilde{h}_t$ are generated during training.

We consider the WGAN framework with the hope that its differentiability properties can help avoid the vanishing gradient and mode collapse problems in the original GAN. We expect the generator to receive helpful gradients even if the dis-
criminators perform well. This approach leads to much better outputs, as discussed in Section 7.

5 Evaluation

5.1 Issues with Existing Methods

In the automatic evaluations of Hu et al. (2017) and Shen et al. (2017), only post-transfer classification accuracy (“Acc”) is considered. That is, a pretrained classifier is used to obtain the classification accuracy of transferred texts. There are potential issues in relying on this metric. First, our empirical results in Section 7 show that training typically converges to a point that gives very poor Acc. Intermediate results are much better in terms of Acc as well as according to manual evaluation and automatic metrics.

Table 1 shows examples of transferred sentences at several points in training using the Yelp dataset. The original sentence has negative sentiment, and the goal is to transfer its sentiment to become positive. We see that high post-transfer accuracy can correlate with poor semantic preservation. This correlation is discussed in more detail in Section 7.

Following similar logic, only considering post-transfer accuracy and semantic preservation can allow disfluent sentences to be considered as good outputs. Only considering post-transfer accuracy and fluency may cause sentences with no relation to the input sentence to be considered as good outputs.

The three metric categories are complementary and help us avoid degenerate behavior in tuning and model selection. At different points throughout training, we have different combinations of accuracy, semantic preservation, and language fluency levels. In general, existing literature has not given details on model selection. We show that intermediate points in training typically lead to generated output that is balanced among the three metrics. We discuss model selection in Sections 5.2 and 8.3.

5.2 Improvements to Evaluation

We now describe three categories of automatic metrics to evaluate textual transfer tasks and study the correlations among them. We then validate the metrics with human judgments in Section 8.2. Our goal is to propose a useful set of metrics that are also unsupervised, i.e., they do not require human-written transferred sentences. This means that our metrics can be used directly for tuning and model selection, even on test data.

Post-transfer Classification Accuracy. The first metric is post-transfer classification accuracy (“Acc”), which is a standard metric used in prior work (Shen et al., 2017). We use a CNN (Kim, 2014) trained to classify a sentence as being from \( X_0 \) or \( X_1 \). This CNN classifier is independent from the discriminators in the textual transfer network. The metric is the percentage of transferred sentences that are classified as belonging to the transferred class.

Semantic Similarity. For the second metric category, we propose two automatic measures of semantic similarity between the original and transferred sentences. The first (“Sim”) is based on cosine similarity of sentence embeddings for the original and transferred sentences. To get sentence embeddings, we average word embeddings weighted by idf weights, defined by \( \text{idf}(t) = \log((|C| - |\{s \in C : t \in s\}| + 1)) \) where \( t \) is a word, \( s \) is a sentence, and \( C = X_0 \cup X_1 \). We use 300-dimensional GloVe word embeddings (Pennington et al., 2014). We average these cosine similarities over all original/transferred sentence pairs. The second similarity metric (“Met”) is the average Meteor score (Denkowski and Lavie, 2014) between the sentences in each original/transferred sentence pair.

Fluency. Transferred sentences can simultaneously exhibit high Acc and Sim/Met, while still being ungrammatical. We therefore add a third automatic metric to target fluency. In particular, we compute perplexity (“PP”) of the transferred corpus. PP is computed using a language model pretrained on the concatenation of \( X_0 \) and \( X_1 \). The language model is separate from the ones used in the style transfer network in Section 4.3. We note that extremely low PP may correspond to nonsensical output that consists solely of common words.

---

4In our context, the generator will then try to generate a sentence as close to the original sentence as possible because of the constraints introduced by the other losses, while trying to fool the discriminator.

5Melnyk et al. (2017) also consider three metrics which correspond to transfer accuracy, semantic preservation, and language fluency. However, no details are provided for how model selection is done.

6On the contrary, letting the program run to convergence will produce very bad results in our experiments, even after extensive parameter tuning.

7We use the embeddings trained on Wikipedia and Gigaword from nlp.stanford.edu/projects/glove/
We thus need to use PP in conjunction with the other two metrics.

Complementarity of Metrics. The examples in Table 1 show that Acc can be inversely correlated with Sim (and Met). Therefore, we plot Sim versus Acc during training. We expect tradeoffs. We want to ask: Under similar post-transfer accuracy, which model gives better semantic preservation? We also examine the plots of PP versus Sim. We want to ask: Given a fixed level of semantic preservation, which model produces more fluent output? We also conduct a manual evaluation and assess the degree to which it correlates with our metrics. More details can be found in Section 8.1.

Summarizing Three Metrics into One Score. It is useful to have a single score that summarizes the three metrics. We propose using an adjusted geometric mean (GM):

$$\text{GM}(t_1, t_2, t_3, t_4) = \min\left(\left\lfloor 100 \cdot \text{Sim} - t_2 \right\rfloor, \left\lfloor 100 \cdot \text{Acc} - t_1 \right\rfloor + \left\lfloor 100 \cdot \text{Acc} - t_1 \right\rfloor_+ \cdot \left(\left\lfloor t_3 - \text{PP} \right\rfloor_+ + \left\lfloor \text{PP} - t_4 \right\rfloor_+ \right) \right)^{\frac{1}{3}}$$

where $[x]_+ = \max(x, 0)$. The thresholds $t_1, t_2, t_3$ can be interpreted as the worst metric values we are willing to accept, in order to balance the weights for the three metrics. We further punish abnormally small perplexities by setting $t_4$, noting that transferred texts with such perplexities typically consist entirely of common words and phrases that do not result in meaningful sentences.

When choosing models, different practitioners may prefer different trade-offs of Acc, Sim, and PP. However, in the absence of any particular use case, we simply choose $t_1 = t_2 = t_3 = 65$ and $t_4 = 0$ in this paper. We expect that, say, 20% increase in $\left\lfloor 100 \cdot \text{Acc} - t_1 \right\rfloor_+$ has similar effects on transfer quality as 20% increase in $\left\lfloor 100 \cdot \text{Sim} - t_2 \right\rfloor_+$, which has similar effects on 20% increase in $\min\{t_3 - \text{PP}\}_+, [\text{PP} - t_4]_+$. To summarize the three metrics into one score, we introduced one simple approach above which we use in this paper. Alternatively, the thresholds could be learned from data if we collect annotations for overall quality of transferred output from a broad range of models. More information is provided in Section 8.

6 Experimental Setup

6.1 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.

Literature. We consider two corpora of literature. The first corpus contains the 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 do not have a large intersection of subject matter, because there are significant differences in terms of the settings, characters, and stories, which leads to harder semantic preservation.

6.2 Hyperparameter Settings

Algorithm 1 requires setting the $\lambda$ weights for each component. Depending on which model is being trained (see Table 2), the $\lambda$’s for the unused losses will be zero. Otherwise, we set $\lambda_1 = 1, \lambda_2 = 0.2, \lambda_3 = 5, \lambda_4 = 0.001, \lambda_5 = 1, \lambda_6 = (0.5)^{ep}$ 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 (Abadi et al., 2015), based on the code by Shen et al. (2017).
6.3 Pretrained Classifiers and Language Models for Evaluation

For the pretrained classifiers, the accuracies on the Yelp and Literature development sets are 0.974 and 0.893, respectively. (Information on train/development/test splits can be found in supplemental materials Section A.1.) For language models, the perplexities on the Yelp and Literature development sets are 27.4 and 49.8, respectively.

7 Results and Analysis

Table 2 shows semantic similarities (Sim and Met) under similar Acc, as well as PP under similar Sim. The first row in each table, M0, is the baseline model of Shen et al. (2017). Figure 1 plots learning trajectories of a subset of the Yelp models shown in Table 2. Models for Literature show similar trends and are plotted in the supplemental material. The figures show trajectories of statistics on corpora generated from the development set during learning. Each two consecutive markers generally deviate by half an epoch of training. Lower-left markers generally precede upper-right markers chronologically during training.

7.1 Analyzing Metric Relationships

The plots of Sim by error rate exhibit positive slopes, meaning that error rate (1 − Acc) is positively correlated with semantic preservation. Curves to the upper-left corner represent better trade-off between error rate and semantic preservation. Our models have curves that reside to the upper left of the baseline M0.

If we observe the plots of PP by Sim, we find that the baseline curve exhibits large positive slope but the curves corresponding to our models do not. This observation indicates that the baseline model sacrifices fluency for semantic preservation. Our models are able to maintain consistent perplexities as similarity increases during training.

The plots also corroborate the fact that we need to look at all three metrics when comparing models. An individual metric on its own cannot effectively determine the quality of a textual transfer system.

7.2 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 the remaining models (M3 to M7).

Paraphrase Loss. Table 2 shows that the model with paraphrase loss (M1) slightly improves semantic similarity over M0 on both datasets under similar Acc. For Yelp, M1 has better Acc and PP than M0 at comparable semantic similarity. Therefore, when used alone, paraphrase loss outperforms the baseline. 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 worse. One challenge in introducing an additional training set of paraphrase pairs is that its notions of paraphrase may clash with those of semantic content preservation given a particular non-parallel transfer dataset. For example, both sets of Yelp reviews share a great deal of semantic content, but the Literature dataset shows systematic differences in semantic content across the two styles, as mentioned in Section 6.1.

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.

Multiple Discriminators. We now compare the trajectories of M0, M2, and M6. For Yelp, Table 2 and Figure 1a show that M6 performs best in terms of semantic preservation. However, M6 converges around an error rate of 0.200, a Sim of 0.825, and a Met of 0.350. We see that M7 performs equally well in semantic preservation, but has much lower perplexity. For Literature, we find that M6 converges around an error rate of 0.225, a Sim of 0.775, and a Met of 0.160. At an error rate of 0.225, no other models perform even close.
Table 2: Results at fixed levels of post-transfer classification accuracy (Acc) and semantic similarity (Sim). Under similar Acc, the best Sim and Met are in bold. Under similar Sim, the best PP is in bold. In both tables, the best GM scores are also in bold. Note that para = paraphrase loss, cyc = cyclic consistency loss, lang = language modeling loss, and 2d = multiple (two sets of) discriminators. Some cells are empty, indicating that the model never reaches the corresponding accuracy or cosine similarity.

Comparing M2 to M6, we see that M6 achieves a better trade-off between Acc and Sim, at a small cost of perplexity. Comparing M5 to M7, M7 achieves a better trade-off. GM is uniformly improved with multiple discriminators. The trends on the Literature dataset are similar, as can be seen in Figures 2a and 2b and Table 2. In summary, if we are seeking lower error rates and are content with slightly less semantic preservation, then we should choose models with multiple discriminators.

Combining Losses. At an Acc of 0.8, M6 and M7 outperform M0 by about 10 points of Sim. At Sim of 0.8, M7 outperforms M0 by about 40 points in PP, while having much larger Acc. We see a similar trend for Literature even though the subject matter differs more between the two styles: At Acc of 0.75, M2 and M6 outperform M0 by 5 points of Sim. At Sim of 0.74, M0 fails to generate meaningful outputs. In order to choose a single model for manual evaluation, we choose M7 for the Yelp dataset and M6 for the Literature dataset as these two models have the highest GM scores.
Table 3: Manual evaluation results (%) using models from left half of Table 2 (i.e., with fixed Acc).

Note: > means “better than”. Each row involves at least 120 untransferred, Model-A-transferred, and Model-B-transferred sentence tuples. A cell is in bold if it represents a model win of at least 10 percentage points.

Table 4: Textual transfer examples, with more provided in Table 6 of the supplemental materials

8 Validating and Improving Metrics by Human Annotation

8.1 Corpus-Level Manual Evaluation

Annotators were shown the original untransferred sentence, as well as sentences produced by two models (which we refer to as Models A and B). They were asked to judge which one of the two better reflects the target style, which one has better semantic preservation of the original, and which one is more fluent. The ordering of the outputs was randomized.

The results of the manual evaluation are shown in Table 3. Overall, the results show the same trends as our automatic evaluation metrics. For Yelp, we find that at similar Acc, M6 and M7 have the best semantic preservation and M7 has the best fluency. For Literature, we find that at similar Acc, M2 has best semantic preservation, and M6 slightly prevails in fluency.

We provide examples in Table 4. They are presented to reflect the general differences among models in an intuitive way.

8.2 Sentence-Level Manual Evaluation: Validation of Individual Metrics

We validate post-transfer classification accuracy by calculating the percentage of machine judgments and human judgments that match. Out of 100 generated sentences for the Yelp dataset, 94 of the machine judgments match with human judgments. For the Literature dataset, 88 judgments match out of 100. Our methods of sampling the generated sentences are described in Section A.3 in the supplemental materials.

We validate the semantic preservation and perplexity metrics by computing sentence-level Spearman’s rank correlation coefficient (ρ) between the metric and human judgments on 100 generated sentences, so as to ensure that the metric is measuring what we desire it to. For Yelp, the Spearman’s ρ between Sim and human ratings of semantic similarity (an integer score from 1 to 4, described in Section A.3 in the supplemental materials) is 0.78. For Literature, the ρ is 0.73.

For Yelp, the Spearman’s ρ between PP and human ratings of fluency (an integer score from 1 to 4, explained in Section A.3) is -0.82. For Literature, the ρ is -0.63. Methods of sampling gener-
ated sentences can be found in Section A.3.

8.3 Improving the Summarizing Metric

The next step is to use human annotations to improve the summarizing metric, as discussed in Section 5.2. In this paper, we used GM as the summarizing metric. Given that each application of textual transfer is different, we may desire different qualities in generated sentences. We can learn a summarizing metric by human annotations in a particular use case of textual transfer. One choice is to still define the metric as \( GM(t_1, t_2, t_3, t_4) \) as in Eq. (7), but then learn the \( t_i \)'s from annotated data. Or, more generally, the metric can be an arbitrary non-linear function \( f(Acc, Sim, PP) \). To obtain annotations to learn such a function, we can ask annotators the general question: “Which of the two transferred sentences is better?” We can thus learn to rank all transferred sentences according to pairwise comparisons (Jamieson and Nowak, 2011; Wauthier et al., 2013; Heckel et al., 2018).

9 Conclusion

We proposed three categories of evaluation 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 and for tuning and model selection. We expect our proposed components to be applicable to a broad range of language generation tasks.

One possible line of future work lies in improving methods for transfer on datasets where the two corpora differ greatly in subject matter, such as the Literature dataset. In such cases, it is difficult to preserve semantics due to the difference in the latent content spaces. It might be helpful, however, to impose explicit restrictions based on external knowledge resources.

Acknowledgements

We thank Karl Stratos and Zewei Chu for helpful discussions, the human annotators for performing manual evaluation, NVIDIA for donating GPUs used in this research, and Google for a faculty research award to K. Gimpel that partially supported this research.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Łukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 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.

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

Nancy Campbell. 1987. Adapted literary texts and the efl reading programme. ELT Journal, 41(2):132–135.

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. arXiv preprint arXiv:1707.02633.

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).

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.

Reinhard Heckel, Max Simchowitz, Kannan Ramchandran, and Martin J Wainwright. 2018. Approximate ranking from pairwise comparisons. arXiv preprint arXiv:1801.01253.

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.

Kevin G Jamieson and Robert Nowak. 2011. Active ranking using pairwise comparisons. In Advances in Neural Information Processing Systems, pages 2240–2248.

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 Lampe, Ludovic Denoyer, and Marc’Aurelio Ranzato. 2017. Unsupervised machine translation using monolingual corpora only. arXiv preprint arXiv:1711.00043.

Jiwei Li, Michel Galley, Chris Brockett, Georgios P Spithourakis, Jianfeng Gao, and Bill Dolan. 2016. A persona-based neural conversation model. arXiv preprint arXiv:1603.06155.

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. arXiv preprint arXiv:1804.06437.

Igor Melnyk, Cicero Nogueira dos Santos, Kahini Wadhawan, Inkt Padhi, and Abhishek Kumar. 2017. Improved neural text attribute transfer with non-parallel data. arXiv preprint arXiv:1711.09395.

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.

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.

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.

Sravana Reddy and Kevin Knight. 2016. Obfuscating gender in social media writing. In Proceedings of the First Workshop on NLP and Computational Social Science, pages 17–26.

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.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. 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.

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.

Fabian Wauthier, Michael Jordan, and Nebojsa Jojic. 2013. Efficient ranking from pairwise comparisons. In International Conference on Machine Learning, pages 109–117.

Tsung-Hsien Wen, David Vandyke, Nikola Mrksic, Milica Gasic, Lina M Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2016. A network-based end-to-end trainable task-oriented dialogue system. arXiv preprint arXiv:1604.04562.

John Wiebing 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, Xuancheng Ren, Xiaodong Zhang, Houfeng Wang, and Wenjie Li. 2018. Unpaired sentiment-to-sentiment translation: A cycled reinforcement learning approach. arXiv preprint arXiv:1805.05181.

Hayahide Yamagishi, Shin Kanouchi, and Takayuki Sato and Mamoru Komachi. 2016. Controlling the voice of a sentence in japanese-to-english neural machine translation. In Proceedings of the 3rd Workshop on Asian Translation (WAT2016), pages 203–210. Association for Computational Linguistics.

Zhen Yang, Wei Chen, Feng Wang, and Bo Xu. 2017. Improving neural machine translation with conditional sequence generative adversarial nets. arXiv preprint arXiv:1703.04887.
Zichao Yang, Zhiting Hu, Chris Dyer, Eric P Xing, and Taylor Berg-Kirkpatrick. 2018. Unsupervised text style transfer using language models as discriminators. arXiv preprint arXiv:1805.11749.

Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2016. Seggan: Sequence generative adversarial nets with policy gradient. arXiv preprint arXiv:1609.05473.

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.

A Supplemental Material

A.1 Dataset Details

Yelp sentiment dataset: 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, respectively. Like Shen et al. (2017), we only use sentences with 15 or fewer words.

Literature dataset: After tokenization, we only use sentences with lengths between 5 and 25. The resulting dataset has 158K, 1.5K, 1.5K Dickens training, development, and testing sentences, respectively, and 162K, 1.5K, 1.5K modern literature sentences, respectively.

A.2 Tables and Plots in Results

Table 5 shows additional results at lower levels of post-transfer accuracy, which show the same trends as the results in Table 2. The plots in Figure 1 show results for additional values of post-transfer accuracy.

Figures 2a and 2b show the learning trajectories for the Literature dataset, which show similar trends as those for Yelp. While the plots for the two datasets appear different from an initial glance, comparing similarities at fixed error rates and comparing perplexities at fixed similarities reveals that the results largely resemble those for the Yelp dataset. The baseline M0 struggles on the Literature dataset. The particularly low perplexity for M0 does not indicate fluent sentences, but rather the piecing together of extremely common words and phrases.

In our analysis, we used Sim as the primary metric for semantic preservation. However, if we were to use Met instead, the plots and our conclusions would be largely unchanged.

A.3 Manual Evaluation Details

We extend our discussion on sentence-level manual evaluation to validate individual metrics. For validation of post-transfer classification accuracy, full details are in the paper. For validation of cosine similarity as semantic similarity, we compute sentence-level Spearman’s rank correlation coefficient between the metric and human judgments on 100 generated sentences.

The metric score is the cosine similarity. The human ranking score has four possibilities (1 to 4). We present pairs of original sentences and transferred sentences to human annotators. They are asked to rate the level of semantic similarity where 1 means “extremely bad”, 2 means “bad/ok/needs improvement”, 3 means “good”, and 4 means “very good.” The human ratings are designed similarly for validating perplexity.

The transferred sentences used in the evaluation are sampled from the development sets produced by models 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.

A.4 Additional Examples

Table 6 provides additional examples.
### Table 5: Results at fixed levels of post-transfer classification accuracy (\(\text{Acc}\)) and cosine similarity (\(\text{Sim}\)). Under similar \(\text{Acc}\), the best \(\text{Sim}\) is in bold. Under similar \(\text{Sim}\), the best \(\text{PP}\) is in bold. The best \(\text{GM}\) scores are also in bold.

| Model        | \(\text{Acc}\) \(\approx 0.800\) | \(\text{Acc}\) \(\approx 0.725\) | \(\text{Sim}\) \(\approx 0.800\) |
|--------------|-----------------------------------|-----------------------------------|-----------------------------------|
|              | \(\text{Acc}\) \(\uparrow\)   | \(\text{Sim}\) \(\uparrow\) | \(\text{PP}\) \(\downarrow\) | \(\text{GM}\) \(\uparrow\) | \(\text{Acc}\) \(\uparrow\) | \(\text{Sim}\) \(\uparrow\) | \(\text{PP}\) \(\downarrow\) | \(\text{GM}\) \(\uparrow\) |
| M0: Shen et al. (2017) | 0.818 0.719 37.3 14.8 | 0.717 0.749 47.2 10.6 | 0.591 0.793 56.1 0.00 |
| M1: +para | 0.819 0.734 26.3 16.6 | 0.725 0.786 30.1 14.5 | 0.704 0.798 31.0 13.5 |
| M2: +cyc | 0.813 0.770 36.4 17.8 | 0.723 0.842 37.4 15.7 | 0.795 0.801 37.4 18.2 |
| M3: +cyc+lang | 0.807 0.796 28.4 18.7 | 0.720 0.837 28.7 15.5 | 0.792 0.802 28.7 18.4 |
| M4: +cyc+para | 0.798 0.783 39.7 17.1 | 0.725 \textbf{0.849} 38.5 \textbf{15.8} | 0.794 0.799 39.4 17.6 |
| M5: +cyc+para+lang | 0.804 0.785 27.1 17.8 | 0.719 0.824 27.4 14.9 | 0.781 0.794 28.0 17.4 |
| M6: +cyc+2d | 0.805 \textbf{0.817} 43.3 17.8 | — — — — | 0.834 0.807 47.7 17.1 |
| M7: +cyc+lang+para+2d | 0.818 0.805 29.0 \textbf{19.6} | — — — — | 0.830 0.799 \textbf{27.8} \textbf{19.5} |

| Literature | \(\text{Acc}\) \(\approx 0.750\) | \(\text{Acc}\) \(\approx 0.675\) | \(\text{Sim}\) \(\approx 0.740\) |
|------------|-----------------------------------|-----------------------------------|-----------------------------------|
|            | \(\text{Acc}\) \(\uparrow\)   | \(\text{Sim}\) \(\uparrow\) | \(\text{PP}\) \(\downarrow\) | \(\text{GM}\) \(\uparrow\) | \(\text{Acc}\) \(\uparrow\) | \(\text{Sim}\) \(\uparrow\) | \(\text{PP}\) \(\downarrow\) | \(\text{GM}\) \(\uparrow\) |
| M0: Shen et al. (2017) | 0.752 0.704 15.3 9.45 | 0.671 0.660 20.1 3.52 | — — — — |
| M1: +para | 0.730 0.713 26.0 10.9 | 0.670 0.730 24.8 7.35 | 0.654 0.733 24.8 4.35 |
| M2: +cyc | 0.748 \textbf{0.751} 38.1 13.9 | 0.675 \textbf{0.783} 50.7 7.81 | 0.739 0.740 40.4 12.5 |
| M3: +cyc+lang | 0.737 0.738 19.8 11.5 | 0.678 0.758 26.5 \textbf{9.30} | 0.715 0.742 \textbf{23.2} 11.2 |
| M4: +cyc+para | 0.741 0.671 28.5 8.17 | 0.670 0.761 38.7 8.36 | 0.705 0.739 30.2 11.4 |
| M5: +cyc+para+lang | 0.735 0.736 24.0 12.1 | 0.676 0.751 27.8 9.00 | 0.720 0.743 27.4 12.1 |
| M6: +cyc+2d | 0.770 0.750 35.7 \textbf{15.2} | — — — — | 0.781 0.740 31.3 \textbf{15.5} |
| M7: +cyc+lang+para+2d | 0.736 0.733 35.9 12.8 | 0.674 0.740 29.2 8.58 | 0.715 0.738 33.4 12.2 |

Figure 2: Learning trajectories with selected models from Table 5. Metrics are computed on the development sets.
| Model | Acc  | Sim  | PP   | GM   | Sentence                                                                 | Style     |
|-------|------|------|------|------|-------------------------------------------------------------------------|-----------|
| Original | —    | —    | —    | —    | they are completely unprofessional and have no experience.              | Negative  |
| M0    | 0.818| 0.719| 37.3 | 14.8 | they are super fresh and well!                                          | Positive  |
| M7    | 0.818| 0.805| 29.0 | 19.6 | they are very professional and have great service.                      | Positive  |
| Original | —    | —    | —    | —    | i would honestly give this place zero stars if i could.                 | Negative  |
| M0    | 0.818| 0.719| 37.3 | 14.8 | i would recommend give this place from everyone again.                  | Positive  |
| M7    | 0.818| 0.805| 29.0 | 19.6 | i would definitely recommend this place all stars if i could.           | Positive  |
| Original | —    | —    | —    | —    | for all those reasons, we won’t go back.                               | Negative  |
| M0    | 0.818| 0.719| 37.3 | 14.8 | for all of pizza, you do you go.                                        | Positive  |
| M7    | 0.818| 0.805| 29.0 | 19.6 | for all those reviews, i highly recommend to go back.                   | Positive  |
| Original | —    | —    | —    | —    | the owner was super nice and welcoming.                                 | Positive  |
| M0    | 0.818| 0.719| 37.3 | 14.8 | the server was extremely bland with all.                                | Negative  |
| M7    | 0.818| 0.805| 29.0 | 19.6 | the owner was very rude and unfriendly.                                 | Negative  |
| Original | —    | —    | —    | —    | this is one of the best hidden gems in phoenix.                         | Positive  |
| M0    | 0.818| 0.719| 37.3 | 14.8 | this is one of the worst restaurants in my life.                        | Negative  |
| M7    | 0.818| 0.805| 29.0 | 19.6 | this is one of the worst restaurants in phoenix.                        | Negative  |
| Original | —    | —    | —    | —    | i declined on their offer, but appreciated the gesture!                 | Positive  |
| M0    | 0.818| 0.719| 37.3 | 14.8 | i asked on their reviews, they are the same time!                       | Negative  |
| M7    | 0.818| 0.805| 29.0 | 19.6 | i paid for the refund, and explained the frustration!                   | Negative  |
| Original | —    | —    | —    | —    | cried the old man, with a dreadful oath.                                | Dickens   |
| M0    | 0.752| 0.704| 15.3 | 9.45 | said the &lt;unk&gt;, &lt;unk&gt;, the &lt;unk&gt;.                        | Modern    |
| M2    | 0.748| 0.751| 38.1 | 13.9 | cried the old man, with a sudden smile.                                  | Modern    |
| M6    | 0.770| 0.750| 35.7 | 15.2 | the old man asked with a deep smile.                                    | Modern    |
| Original | —    | —    | —    | —    | something that ’ll do you good service, young feller,                   | Dickens   |
| M0    | 0.752| 0.704| 15.3 | 9.45 | that ’s do n’t know, she said.                                         | Modern    |
| M2    | 0.748| 0.751| 38.1 | 13.9 | well that you do you ’re right, vincent.                                | Modern    |
| M6    | 0.770| 0.750| 35.7 | 15.2 | perhaps that will do you good, man, said my father.                    | Modern    |
| Original | —    | —    | —    | —    | she asks him to help her.                                               | Modern    |
| M0    | 0.752| 0.704| 15.3 | 9.45 | she was not to be &lt;unk&gt;.                                           | Dickens   |
| M2    | 0.748| 0.751| 38.1 | 13.9 | she says him to give her.                                               | Dickens   |
| M6    | 0.770| 0.750| 35.7 | 15.2 | she begged him to take her.                                             | Dickens   |
| Original | —    | —    | —    | —    | gina ’s face lit-up in delight.                                         | Modern    |
| M0    | 0.752| 0.704| 15.3 | 9.45 | they ’s &lt;unk&gt; in her.                                               | Dickens   |
| M2    | 0.748| 0.751| 38.1 | 13.9 | scrooge ’s voice comes in astonishment.                                 | Dickens   |
| M6    | 0.770| 0.750| 35.7 | 15.2 | says mr. pickwick allen in astonishment.                                | Dickens   |

Table 6: More textual transfer examples