Optimizing BLEU Scores for Improving Text Generation

Terrance Liu (terrancl) [1]

Abstract

In typical text generation settings, there exists a discrepancy between the training objective and evaluation criteria. Generally, researchers train text generation models by maximizing for log-likelihood on cross-entropy while evaluating on a separate metric, such as BLEU score, that cannot be optimized by evaluating gradients. While some work has been done to use policy gradients to directly optimize for non-differentiable rewards, these methods are difficult to train due to the amount of computation required. Instead, we focus on an alternative method of approximating for BLEU by evaluating Differentiable Expected BLEU (DEBLEU). We evaluate our method on machine translation, while comparing results to both cross-entropy and policy gradient methods. Furthermore, we provide empirical results regarding the computational benefits of DEBLEU vs. policy gradients. Lastly, we focus on text-style transfer and argue that DEBLEU loss is a more suitable objective than cross-entropy for this task.

1. Introduction

In recent years, the natural language processing community has developed growing interest in text generation. Strong improvements have been made across a wide variety of tasks, such as machine translation and dialog systems. In general, researchers have tackled text generations tasks by maximizing log-likelihood during training. However when actually evaluating the quality of their experiments, researchers use metrics such as BLEU score (Papineni et al., 2002). Because of the popularity of BLEU scores in NLP, addressing this inconsistency between the training objective and evaluation metric is a well-researched area, and finding a solution can have significant impact on a wide variety of tasks popular today.

For our project, we explore ways of directly optimizing for BLEU scores in the context of text generation. Therefore, the main issue we need to overcome is the fact that BLEU is a non-differentiable function, making it impossible to apply gradient descent. The main problem we wish to tackle then is to explore methods of approximating this objective.

Currently, there exists some work that attempts to address the same issue. Borrowing techniques from reinforcement learning, Ranzato et al. uses a policy gradient with the BLEU score as a reward to optimize for this metric for text generation. However, this leads to a major challenge, as the sampling required for this technique is expensive and difficult. Moreover, the gradient estimation itself suffers from high variance. Instead, we would like take an alternative approach of making soft approximations to the n-gram matching counts that is used to calculate BLEU scores. In doing so, we use a differentiable approximation, DEBLEU, and compare results to both policy gradient methods and state-of-the-art techniques that maximize log-likelihood (Wang et al., 2019).

In our project, we would like to study the advantages of directly optimizing for BLEU scores rather than log-likelihood in the context of text style transfer. In our experiments, we first show that approximating BLEU using DEBLEU is computationally more efficient than using policy gradient methods. Second, we test that DEBLEU is resilient to perturbations in the ground truth sentence when compared to cross-entropy loss, which can be very sensitive to adding, deleting, or replacing tokens. Finally, we demonstrate that this property is useful in the context of text style transfer tasks, where one may try to modify characteristics such as sentiment. In such cases where modifying style may only require changes to a few tokens, optimizing for DEBLEU loss is preferable to maximum likehood estimation.

2. Background and Related Work

2.1. BLEU

We first discuss the BLEU score, which is a common metric that researchers use to evaluate natural language processing systems (Papineni et al., 2002). Papineni et al. define BLEU as a weighted geometric mean of n-gram precision scores, defined as

[1] Carnegie Mellon University, Pittsburgh, PA 15213, USA.

10708 Class Project, Spring, 2019.

$$\text{prec}_n = \frac{\sum_{\mathbf{s}} \min(C(\mathbf{s},\mathbf{y}), C(\mathbf{s},\mathbf{y}^*))}{\sum_{\mathbf{s}} C(\mathbf{s},\mathbf{y})}$$

where $\mathbf{y}$ is the hypothesis sequence, $\mathbf{y}^*$ is the ground-truth sequence, $\mathbf{s}$ is an n-gram subsequence of $\mathbf{y}$, and $C(\mathbf{s}, \mathbf{y})$ is the number of times $\mathbf{s}$ appears in $\mathbf{y}$. Then BLEU is formulated as

$$\text{BLEU} = \text{BP} \exp\left(\sum_{n=1}^{N} w_n \log \text{prec}_n\right)$$

BP stands for brevity penalty, which penalizes sequences that are too short. N determines how many n-gram precision-scores to use. Finally, w_n is the weight of each precision score, which often is set to $1/N$.

$$\text{BP} = \begin{cases} 1 & T > T^* \\ e^{(1-r/c)} & T \leq T^* \end{cases}$$

Where T is the prediction sequence length and T^* is the reference sequence length.

2.2. Optimizing BLEU

Ranzato et al. directly optimizes for their final evaluation metric by borrowing ideas from reinforcement learning. Their proposed algorithm, Mixed Incremental Cross-Entropy Reinforce (MIXER) is adapted from the REINFORCE algorithm (Williams, 1992), which is suitable to this problem since it does not require rewards to be differentiable. In their work, Ranzato et al. treat the text generation model as an *agent*, the parameters of model as the *policy*, and each generated word as an *action*. Then at the end of each generated sequence, MIXER calculates the BLEU score as the observed *reward*.

However, one major challenge of using the policy gradient is that the action space for text generation is very large, making training very difficult. Therefore to ameliorate the problems of convergence, MIXER trains an RNN with cross-entropy for the task and uses this model as it's initial policy, rather than beginning with a random policy. In addition, MIXER employs an annealing schedule, which trains the model to generate more stable sequences.

More recently, researchers have tackled this issue by making BLEU differentiable. Recognizing that sampling procedure in reinforcement learning methods like MIXER are computationally expensive, Zhukov et al. proposes optimizing for the lower bound of expected BLEU score. Through a set of assumptions and derivations, Zhukov et al. derive an expression that coincides with the exact value for an individual n-gram matching score. Taking the product of these for the aggregate n-gram scores used to calculate BLEU, they are able to derive a lower bound that is differentiable. They then run their method on toy tasks and small translation tasks to make comparisons to reinforcement learning approaches. Similarly, Casas et al. derive a differentiable version of BLEU by making an assumption that hypothesis and ground truth sequences have the same length. From this, they follow a series of matrix computations to calculate the BLEU score.

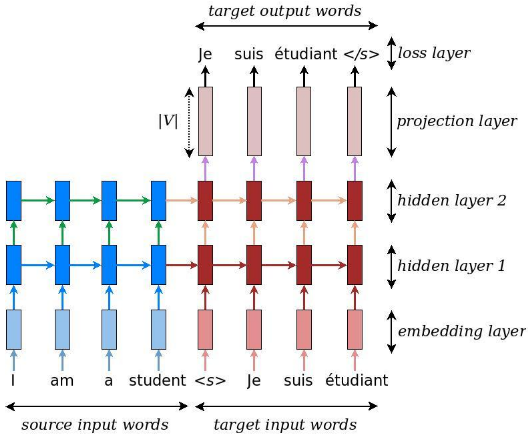

Figure 1. Diagram of a sequence-to-sequence architecture, which uses a unidirectional, multi-layer RNN with an LSTM recurrent unit. In this example, the model translates a source sentence ”I am a student” into a target sentence ”Je suis tudiant” (Luong et al., 2017).

In addition, Yavuz et al. also attempt to solve this objective discrepancy problem between MLE and BLEU scores, but rather than optimizing for BLEU itself, they introduce a new objective (CaLcs) that also captures sequence level structure similarity. This objective is an approximation of the longest subsequence (LCS) metric, and rather than completely replace log-likelihood, they add it on top as an additional objective for text generation models by pretraining using cross-entropy and continue training using the CaLcs objective. Having bounded the approximation error of LCS using CaLcs in their paper, Yavuz et al. then run several experiments in abstractive summarization and machine translation, achieving increases in BLEU scores.

2.3. Machine Translation

Popular among machine translation tasks are sequence-to-sequence networks, which uses an encoder-decoder framework (Sutskever et al., 2014). This method uses a multilayered LSTM to map an input sequence to a vector and then another LSTM to decode this vector into the target sequence. See Figure 1. Following this work, Bahdanau et al. proposed a new method that adds attention to the sequence-to-sequence model, having posited that the fixed-length vector was causing a bottleneck. They showed that such soft-alignments improves performance. More recently, researchers have been able to achieve state-of-the-art results in machine translation using Transformers, which forgo recurrent and convolutional neural network architectures for

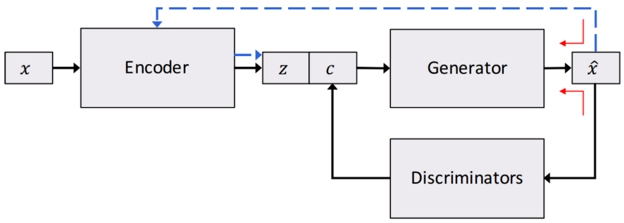

Figure 2. **z** is the unstructured latent code **c** is the structured code for sentence attributes to be controlled (Hu et al., 2017).

one based solely on attention mechanisms (Vaswani et al., 2017).

2.4. Text Style Transfer

As stated previously, our main area of focus is text style transfer, which we hypothesize that DEBLEU is more suitable for since it is less sensitive to token perturbations than maximum likelihood estimation. We therefore hope to run experiments adapted from recent text style transfer techniques (Hu et al., 2017; Yang et al., 2018). Hu et al. leverages VAEs for text generation. Using differentiable approximations to discrete text examples, they are able to define explicit constraints on attribute controls and use VAEs with style discriminators to learn interpretable sentence representations. Specifically, they show that they are able to control sentiment and tense while generating realistic sentence samples. Figure 2 provides a high-level diagram of the architecture Hu et al. introduces.

For the generator, we have the loss $\mathcal{L}_G = \mathcal{L}_{VAE} + \lambda_c \mathcal{L}_{Attr,c} + \lambda_z \mathcal{L}_{Attr,z}$. $\mathcal{L}_{VAE}$ is the standard VAE loss that minimizes cross-entropy. Given that **c** is your attribute code (for example, sentiment label), **z** is the latent variable learned by the VAE, and $\tilde{G}_\tau(\mathbf{z}, \mathbf{c})$ is the soft generated sentence using Gumbel-softmax decoding (Jang et al., 2016), we have then that

$$\mathcal{L}_{Attr,\mathbf{c}} = \mathbb{E} \log q_D(\mathbf{c} | \tilde{G}_\tau(\mathbf{z}, \mathbf{c}))$$
$$\mathcal{L}_{Attr,\mathbf{z}} = \mathbb{E} \log q_E(\mathbf{z} | \tilde{G}_\tau(\mathbf{z}, \mathbf{c})$$

For the discriminator, we have the loss $\mathcal{L}_D = \mathcal{L}_s + \lambda_u \mathcal{L}_u$, where $\mathcal{L}_s$ is the cross-entropy loss for predicting the sentiment code using the input sentence and $\mathcal{L}_u$ is the cross-entropy loss using noisy synthesized sentence-attribute pairs generated by the model. Note that λ_c, λ_z, and λ_u are all hyper-parameters used to weight each loss.

Finally given the above objectives, Hu et al. alternate between training the discriminator and generator until reaching convergence.

Yang et al. tackles unsupervised text transfer using a different approach. While Hu et al. uses a style classifier, Yang et al. leverages adversarial training with a binary classifier. However, unlike previous approaches that use a binary discriminator, Yang et al. train a language model to that assigns a probability of how likely a sentence is real. Empirically, they are able to achieve good results while eliminating the need for negative samples, thereby stabilizing training. In the context of sentence manipulation, they are able to achieve better accuracy and language perplexity scores when compared to the results in Hu et al..

3. Methods

3.1. DEBLEU

We first adapt the Differentiable Expected BLEU (DEBLEU) objective for our experiments (Wang et al., 2019). In their work, Wang et al. leverages the sparsity of the standard BLEU metric to derive an approximation that is differentiable. Overall, we believe that optimizing this metric will improve performance since we are directly optimizing for BLEU score rather than some proxy (i.e. cross-entropy). In addition, we expect faster training time since calculating DEBLEU loss does not require sampling needed in reinforcement learning (i.e. the policy gradient method).

Following a common approach used to for the policy gradient algorithm (Ranzato et al., 2015), Wang et al. begin with the expected BLEU objective:

$$\mathcal{L} = \mathbb{E}_{p_\theta(\mathbf{y})}[\text{BLEU}(\mathbf{y}, \mathbf{y}^*)]$$

Subsequently, Wang et al. approximate the expectation as in (Zhukov et al., 2017):

$$\mathcal{L} \approx \text{BP} \prod_{n=1}^{N} (\mathbb{E}_{p_\theta(\mathbf{y})} \,\text{prec}_n)^{w_n}$$

where prec_n is the n-gram precision defined for the traditional BLEU score metric. Leveraging sparsity properties of the n-gram precision, they approximate $\mathbb{E}_{p_\theta(\mathbf{y})} \,\text{prec}_n$, which we denote as $\widetilde{\text{prec}}_n$.

$$\widetilde{\text{prec}}_n = \frac{1}{T-n+1} \sum_{i=1}^{T-n+1} \tilde{o}_{n,i}$$

where $\tilde{o}_{n,i}$ is defined as

$$\tilde{o}_{n,i} = \min\left(1, \frac{v^*_{n,i}}{\mathbb{E}_{p(y_{\neg i:i+n})} v_{n,i}}\right)$$

Note: in the following notation, Wang et al. enumerate over token indices rather than use the conventional n-gram formulation. Figure 3 provides a pictoral representation of this indexing.

$$C\left(y_{i:i+n}, y\right) = \sum_{i'=1}^{T-n+1} \mathbf{1}\left[y_{i':i'+n} = y_{i:i+n}\right] \triangleq v_{n,i}$$
$$\text{C}(y_{i:i+n}, y^*) = \sum_{j'=1}^{T^*-n+1} \mathbf{1}\left[y^*_{j':j'+n} = y_{i:i+n}\right] \triangleq v^*_{n,i}$$

To evaluate $v_{n,i}$ and $v^*_{n,i}$, Wang et al. consider 3 cases

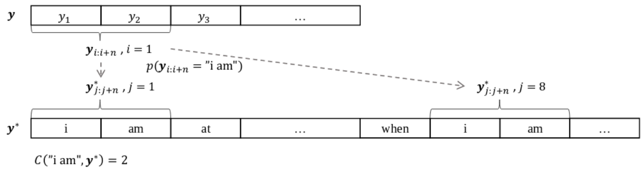
$\boldsymbol{y}$
y_1
y_2
y_3
...
$\boldsymbol{y}_{i:i+n}$, $i = 1$
$p(\boldsymbol{y}_{i:i+n}$ = "i am")
$\boldsymbol{y}^*_{j:j+n}$, $j = 1$
$\boldsymbol{y}^*_{j:j+n}$, $j = 8$
$\boldsymbol{y}^*$
i
am
at
...
when
i
am
...
C("i am", $\boldsymbol{y}^*$) = 2

Figure 3. An example value of $y_{i:i+n}$. Here, $i = 1$ and $n = 2$, where $y_{i,i+n}$ takes value of i am, which occurs twice in y (at $j = 1$ and $j = 8$) (Wang et al., 2019).

when comparing $y_{i':i'+n}$ and $y_{i:i+n}$ to calculate $u_{i,j} = \mathbb{E}_{p(y_{\neg i,i+n})} \mathbf{1}\left[y_{i':i'+n} = y^*_{j:j+n}\right]$

1. The two refer to the same n-gram (i.e. $i' = i$):

$$u_{i,j} = 1$$

2. The two do no overlap (i.e., $|i' - i| \geq n$):

$$u_{i,j} = p\left(y_{i':i'+n} = y^*_{j:j+n}\right)$$

3. The two overlap (i.e. $0 < |i' - i| < n$):

$$u_{i,j} \geq p\left(y_{i':i'+n} = y^*_{j:j+n}\right)$$

For case 3, we approximate $u_{i,j} \approx p\left(y_{i':i'+n} = y^*_{j:j+n}\right)$, giving us

$$\sum_{j=1}^{T^*-n+1} \min\left(1, \frac{p_\theta\left(y_{i::+n}=y^*_{j:j+n}\right)}{1+\sum_{\substack{i'=1\\ i\neq i}}^{T-n+1} p_\theta\left(y_{i':i'+n}=y^*_{j:j+n}\right)}\right)$$

Finally taking the *logarithm*, we arrive at our loss function:

$$\mathcal{L}_{\text{DEBLEU}} = -\log \text{BP} - \sum_{n=1}^{N} w_n \log \widetilde{\text{prec}}_n$$

3.2. Teacher Masks

As in (Hu et al., 2017; Wang et al., 2019), we resolve the issue of backpropgating through discrete samples by using Gumbel-softmax decoding (Jang et al., 2016). This approach produces soft inputs for each sampled token, which we use to calculate the DEBLEU loss and its gradients. In practice however, Wang et al. found that replacing the hard counts with probabilities used as soft counts generated errors that accumulated over each token and caused instability in the training.

To address this issue, Wang et al. introduce ”teacher masks”, in which the soft input distribution $p_\theta(y_i, y_{1:i})$ is replaced with the one hot representation of the ground-truth token (see Figure 4). In their experiments, they gradually annealed this mask pattern by increasing the proportion of unmasked steps until eventually removing the masks altogether (gradually increasing the difficulty of the optimization problem). In our experiments, we use a (2:2) and (4:2) mask pattern, which we remove after a set number of epochs.

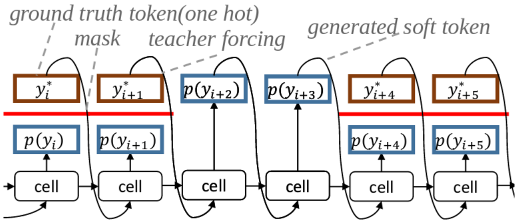

Figure 4. Depicts a 2:2 mask pattern, in which red lines represent masked steps (in which one-hot ground truth tokens are used). Otherwise, a Gumbel-softmax distribution output is used as a soft token (Wang et al., 2019)

.

3.3. Controllable Text Generation

We provide a high-level overview for our style-transfer task methodology. Our model follows the architecture in (Hu et al., 2017) with some simplifications. First, we do not use samples generated by the decoder to train our model. Thus, we drop the terms $\lambda_z \mathcal{L}_{Attr,z}$ and $\lambda_u \mathcal{L}_u$ from the losses $\mathcal{L}_G$ and $\mathcal{L}_D$ respectively. Next, we a use an RNN encoder and an attentional RNN Decoder for our generator. Following Yang et al., we evaluate the generated sentences on BLEU score, as opposed to just the attribute code prediction accuracy. This evaluation serves to measure how realistic the generated sentences are and how similar they are to the original inputs.

4. Experiments

We first present the key questions that we wish to answer using the results of our experiments.

1. Does optimizing DEBLEU loss improve performance compared to cross-entropy optimization and using a policy gradient?

2. Does optimizing DEBLEU loss require less computation (which we measure with training time)?

3. Is DEBLEU loss less sensitive to sequence perturbations when compared to cross-entropy?

4. If so, does this property of DEBLEU loss translate well to text style transfer tasks?

Method	de-en	en-fr
Cross-entropy	22.78	38.13
Policy Gradient	23.35	39.45
DEBLEU	**24.12**	**39.65**

Table 1. BLEU score results for neural machine translation tasks on the test sets, averaged over 5 runs. In our baseline model, we use cross-entropy loss and compare it to the policy gradient method and to our method using DEBLEU loss. We evaluate on the IWSLT14 German-to-English (de-en) and the English-to-French (en-fr).

4.1. Machine Translation

4.1.1. Results

We first replicate results in (Wang et al., 2019), using a sequence-to-sequence model with attention as our base model, pretraining using cross-entropy loss and then further training using our approximated DEBLEU objective. Beam search is used for decoding. We also compare our results to fine-tuning with the policy gradient, rather than DEBELU. For our dataset, we use German-to-English (de-en) IWSLT14 and English-to-French (en-fr) (Cettolo et al., 2014). We follow a (2:2), (4:2), (1:0) teacher-mask annealing schedule for de-en and (2:2), (8:2), (1:0) schedule for en-fr.

As seen in Table 1, we reproduce results from (Wang et al., 2019), achieving an improvement over the baseline of **1.34** and **1.52** on the de-en and en-fr ISWLT14 datasets respectively. We also see slight improvements in comparison to the policy gradient method, reporting BLEU score improvements of **0.77** and **0.20** on our two datasets.

In addition, we evaluate computational complexity empirically by comparing training time for the policy gradient and DEBLEU methods. We find that optimizing for DEBLEU loss is more computationally efficient than using a policy gradient. We report a **2.14x** speedup for machine translation tasks.

4.1.2. Discussion

We provide analysis on the both performance and computational complexity of optimizing for DEBLEU. The main result is that fine-tuning using DEBLEU loss can provide a noticeable improvement, when compared to training only with cross-entropy. In addition in our experiments, the DEBLEU provides a slight improvement over using policy gradient. One may argue that the improvement is not significant, especially when evaluated on en-fr. However, our results still suggest that DEBLEU seems to perform at least as well as reinforcement learning methods. In addition, the speedup in training time is a clear advantage over policy gradient. Overall, our experiments provides evidence that if we wish to optimize for BLEU score directly, we should consider using DEBLEU since it provides similar improvements when compared to the policy gradient method but with less computation. Moreover, even if we did not care about training time, it would still be worthwhile to try both methods since they are comparable in performance.

For further work regarding optimizing for DEBLEU loss in neural machine translation, we would like to modify the experiment by testing on a larger dataset, namely the English-to-German (en-de) WMT14 dataset. In addition, rather than using a sequence-to-sequence model, we also use transformer networks to evaluate our DEBLEU objective (Vaswani et al., 2017). Although we report positive results here, we can provide a much more convincing argument for DEBLEU by evaluating on a more difficult task and using state-of-the-art methods.

4.1.3. Caveats and Limitations

Examining Figure 5, we can clearly see the improve that optimizing DEBLEU loss has over training with just cross-entropy and using the policy gradient method. However, there are some caveats that one can point out.

First, we notice that the policy-gradient method performance drops fairly quickly. On one hand, this is a clear disadvantage. However, one could argue that this negates the computational advantage that optimizing for DEBLEU loss has since although each step takes almost twice as long when using the policy gradient method, the method does not require one to train for as many steps. However, we argue that the performance increase, especially on the de-en dataset, is still a major advantage.

Next, the plot of the test BLEU score on the en-fr dataset in Figure 5 shows that the performance plateaus after annealing from a (2:2) mask to (8:2). This observation suggests that the teacher masks are somewhat fragile. However, a positive takeaway is that the experiment suggests that we may be able to push performance further. Perhaps the (2,2), (8:2), (1:0) is not optimal for this dataset. We hope in future works to experiment with additional annealing schedules to improve the performance on en-fr, which would greatly benefit our results since currently on our own experiments, the difference between DEBLEU and policy gradient is not signficiant on the English-to-French task.

4.2. Perturbation Studies

We first run experiments to demonstrate how modifying tokens in the ground-truth sentence affects BLEU score, cross-entropy loss, and DEBLEU loss. In this experiment, we train a RNN encoder-decoder model on the Yelp sentiment dataset, optimizing for cross-entropy loss. We train for 10 epochs until the neural network can almost perfectly

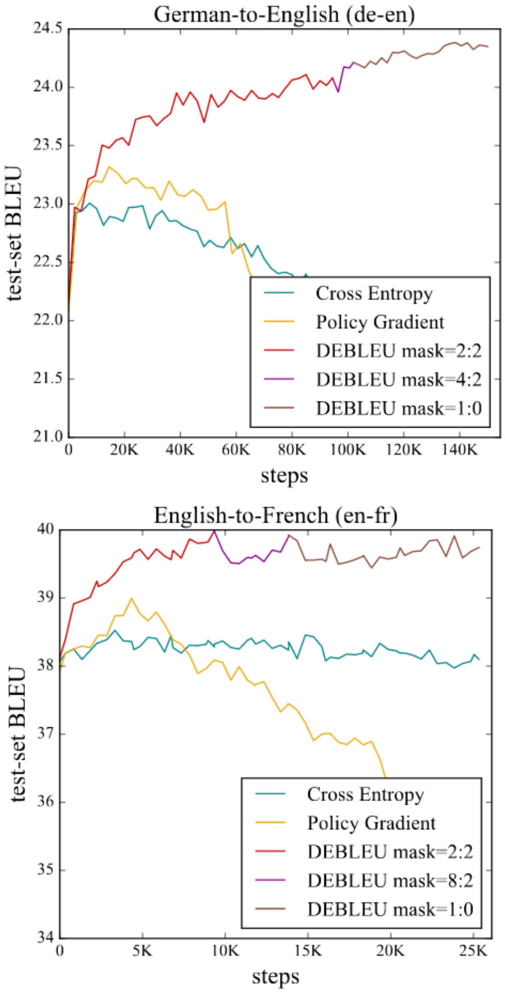

Figure 5. Test-set BLEU score when training (after pretraining using cross-entropy) on ISWLT14 German-to-English (de-en) and English-to-French (en-fr) datasets. Graphs are taken directly from (Wang et al., 2019), but are indicative of the experiments we have replicated.

reconstruct the input sentences. We achieve a BLEU score of 99.908 on the test set.

Next, we run experiments in which we randomly perturb tokens in the ground-truth sentence and evaluate the autoencoder's outputs using the above metrics. For replacement experiments, we randomly pick a token and replace it with an unknown tag outside of the embedder vocabulary.

As seen in the Figure 6, increasing the number of tokens we replace or delete decreases the BLEU score dramatically. Moreover, we observe that replacing and deleting tokens from the ground-truth sentence roughly have the same effect on the BLEU score metric.

In Figure 7, we plot the effects of replacing and deleting tokens on cross-entropy and DEBLEU loss metrics. We

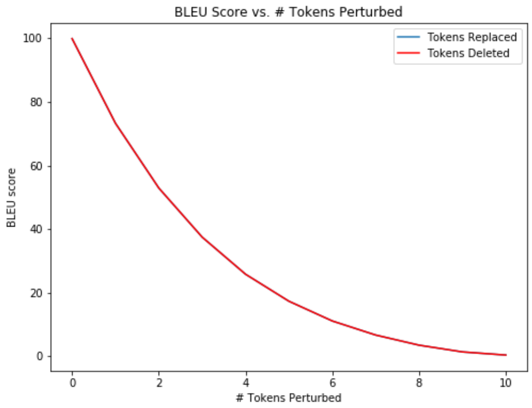

Figure 6. The curves for the affect on the BLEU score of perturbing ground-truth sentences.

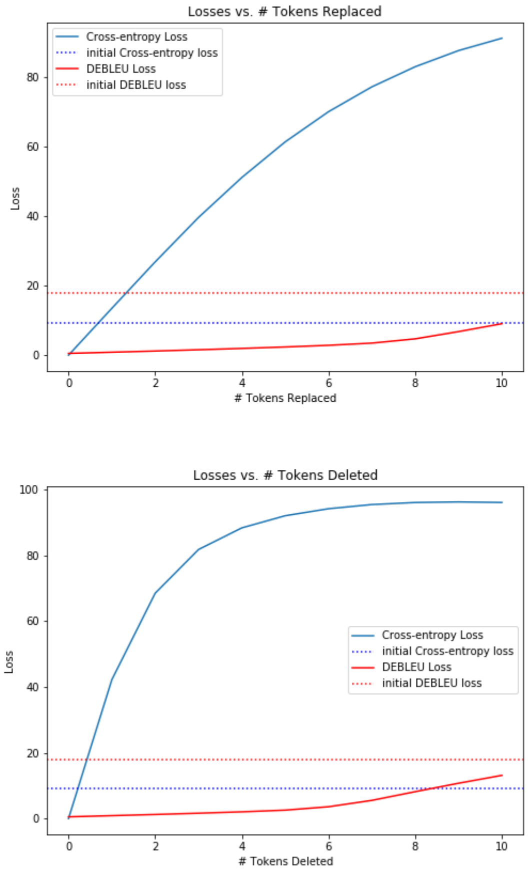

Figure 7. The curves for the affect on the cross-entropy and DEBLEU loss score of replacing and deleting tokens in ground-truth sentences. Initial (prior to training the autoencoder) cross-entropy and DEBLEU loss metrics are plotted as dotted lines.

first notice that cross-entropy loss is much more sensitive to deleting tokens than to replacing, even though the two perturbations have the same effect on BLEU score. The cross-entropy loss spikes up dramatically, even after deleting just 1 or 2 tokens. We also confirm our hypothesis that in both cases, cross-entropy loss is more sensitive than the DEBLEU loss metric. To further emphasize this effect, we plot the initial cross-entropy and DEBLEU losses before training to get a sense of how much these values change when perturbing the ground-truth sentences. We can see that the cross-entropy loss far exceeds the loss incurred when the model has not seen any data, highlighting how quickly this loss can blow up. DEBLEU loss however never exceeds the initial loss value.

Overall, this experiment gives us confidence that DEBLEU loss may be more appropriate for style-transfer text generation. In a sentence such as *The food is good*, one can change the sentiment from positive to negative by just replacing the token *good* with *bad*. However, as shown in our experiments, cross-entropy loss is sensitive to replacing even one token.

4.3. Text Style Transfer

4.3.1. Setup

In section 3.3, we gave a high-level overview of how we adapt the architecture presented in (Hu et al., 2017) for our style transfer experiments. We provide further details on the experimental setup, which uses the Yelp sentiment dataset.

For our baseline model, we following (Hu et al., 2017) and pretrain for 10 epochs in which $\lambda_c = 0$, essentially forcing the generator to generate proper sentences before training it to infer the attribute code. We then train for an additional 2 epochs, setting $\lambda_c = 0.1$, effectively training the model to classify decoded sentences correctly.

To test the effectiveness of our DEBLEU loss, we run an experiment similar to the baseline experiment described above. However after the initial 10 epochs of pretraining, we add our DEBLEU loss to the generator loss, giving us

$$\mathcal{L}_G = \mathcal{L}_{VAE} + \lambda_D \mathcal{L}_{DEBLEU} + \lambda_c \mathcal{L}_{Attr,c}$$

Where λ_D is a tunable parameter. In our experiments, we choose values from $\{0, 0.1, 0.5, 1.0\}$ for λ_D (with $\lambda_D = 0$ corresponding to the baseline). We begin optimizing for the DEBLEU loss using an initial (2:2) teacher mask, switching to a (4:2) mask after 2 epochs. Finally, we train without a mask, i.e. (1:0), for an additional epoch. Beam search is used for decoding.

	BLEU	**Accuracy**
$\lambda_D = 0$ (baseline)	53.63	90.75
$\lambda_D = 0.1$	**55.45**	90.61
$\lambda_D = 0.5$	52.12	91.12
$\lambda_D = 1.0$	53.07	89.83

Table 2. Test-set results for the text-style transfer tasks. We report the sentiment classification accuracy of the decoded sentences as well as the BLEU score when evaluated on the ground truth sentences. Experiments were run on the Yelp sentiment dataset.

Flipping sentiment code	
very friendly guys	very unhappy guys
a fun, eclectic eatery	a crap, pathetic eatery
great food, friendly staff	pathetic food, unhappy staff
terrible experience	delicious experience
they did a fabulous job!	they did a crappy job!

Table 3. Hand-picked samples from the test set for sentiment transfer on the Yelp sentiment set. We use results from our experiment where $\lambda_D = 0.1$.

4.3.2. Results and Discussion

We evaluate our experiments using the BLEU score with the original sentence and the sentiment classification accuracy. Results are reported in Table 2. We achieve a BLEU score improvement of **1.82** by further optimizing the DEBLEU loss, while maintaining a nearly identical binary classification accuracy. In Table 3, we provide a few shorter, hand-picked examples that display the effectiveness of the model.

Our experiment demonstrated that optimizing for BLEU score using DEBLEU loss is effective for text-style transfer. The positive results when compared to the baseline of only using cross-entropy indicates supports are hypothesis formed after conducting the perturbation studies. Cross-entropy is more sensitive to sequence token changes, which makes it less suitable for a task in which we want to encourage some differences between the generated sentence and the ground truth. Moreover, we did not have the resources to do thorough hyperparameter tuning on the three loss components. We hope to evaluate on more values of λ_D. In addition, we held λ_c constant, even though ideally, we would wish to tune this parameter as well to maximize performance.

4.3.3. Caveats and Limitations

Despite these positive results however, there are several caveats that we wish to address regarding the results of our experiments. One major concern is that it is unclear how significant a 1.82 increase in BLEU score is. Although on some tasks such as machine translation, this would amount to a large increase in performance over state-of-the-art, we

Flipping sentiment code	
service was great	worthless was pathetic
so good!	so rant!
what a treat!	what a refuse!

Table 4. Hand-picked samples from the test set for sentiment transfer on the Yelp sentiment set. We specifically choose examples that are of lower quality to demonstrate a limitation of the text-style transfer model. We use results from our experiment where $\lambda_D = 0.1$.

cannot conclude the same on this task.

First, a BLEU score of 53.63 is already fairly high and although 55.45 is a noticable improvement, it is unclear how valuable this increase is, especially given that increases BLEU score do not perfectly correlate to higher quality sentences based on human judgment. Perhaps a stronger improvement in BLEU score could add confidence to the effectiveness of optimizing for DEBLEU loss. However, as seen in Figure 6, modifying just a few tokens will lower the BLEU score significantly. If a score of 55.45 is already near the maximum value we could achieve for this task, it may be difficult to push this metric further and perhaps we need to focus more on other parts of the problem, such as improving accuracy.

Second, for a task such as style-transfer, it is difficult to measure how important BLEU score, since there are multiple metrics that matter for this problem. Perhaps the marginal benefit of increasing BLEU score past 53.63 is very low and that a higher accuracy is more important. Perplexity is also an important metric. Although a higher BLEU score would prevent the model from outputting nonsense, such as just outputting the word *"bad"* whenever we want a sentence with negative sentiment, language perplexity is still important for generating realistic sentences. For example, in Table 4, we list examples in which the generated sentence has the correct sentiment but low perplexity. When compared to those in Table 3, these samples appear awkward and are of lower quality. Therefore, although we have achieved a higher BLEU score, it is difficult to measure how much better we have done on this task. One possible way to measure this is to have humans manually compare generated sentences, but this is a very costly endeavor.

An additional experiment that could add insight into this issue is the incorporate a language model as a discriminator for this model. As stated earlier in this report, Yang et al. run experiments by modifying the model from (Hu et al., 2017). Empirically, they show that they can achieve much better language perplexity while maintaining a similar level of classification accuracy. However in the process of decreasing language perplexity, the model also achieves a lower BLEU score than Hu et al.'s base model. An interesting result would be to try to maintain similar levels of accuracy and language perplexity while improving BLEU score of the generated sentences. This setting could contain an added advantage as well since the BLEU score using just cross-entropy is lower, leaving possibly more room for improvement.

4.3.4. Training Instability

Finally, we observed some instability during the training process when optimizing for DEBLEU loss. The DEBLEU had some trouble converging when run for longer epochs, possibly due to the model also optimizing for classification loss. The evaluated BLEU score also jumped around between epochs, reaching as low as 30 before climbing back up to the 50s in the last two epochs.

There are a few options for future experiments that are worth investigating. The first is that we would like to explore different teacher masks. We tried different combinations of (4:4), (2:2), (4:2), (16:2), (8:2), and (1,0). In the end, we settled on the (2:2), (4:2), (1:0) schedule since the other options did not seem to make a difference. In addition, we currently anneal based on the number of epochs, but perhaps it would make sense to a convergence trigger to determine when to switch between masks. The main deterrence is that after pretraining, we decrease both DEBLEU loss and classification loss, which unsurprisingly compete with each other. It is unclear what the validation metric should be for using as a convergence trigger.

5. Conclusion

We have explored the differentiable expected BLEU objective in the context of text generation, addressing discrepancy issues between training (using cross-entropy) and evaluation (using BLEU score). The method has the additional benefit of being computational more efficient that policy gradient methods that require sampling during training. We demonstrate the usefulness of this metric, having run experiments on machine translation. Moreover, we focus on text-style transfer, having shown that DEBLEU is less sensitive to token perturbations than is cross-entropy loss. This added benefit makes the objective suitable for sentiment-transfer text generation, which we observe in our experiments as well. For future work, we believe that running the additional experiments described in discussion subsections of section 4 would be very beneficial. In addition, we hope to find additional tasks in which this metric can be useful, ideally improving upon state-of-the-art results while limiting increases in computation.

References

Bahdanau, D., Cho, K., and Bengio, Y. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

Casas, N., Rodríguez Fonollosa, J. A., and Ruiz Costa-Jussà, M. A differentiable bleu loss. analysis and first results. In *ICLR 2018 Workshop Track: 6th International Conference on Learning Representations: Vancouver Convention Center, Vancouver, BC, Canada: April 30-May 3, 2018*, 2018.

Cettolo, M., Niehues, J., Stüker, S., Bentivogli, L., and Federico, M. Report on the 11th iwslt evaluation campaign, iwslt 2014. In *Proceedings of the International Workshop on Spoken Language Translation, Hanoi, Vietnam*, pp. 57, 2014.

Hu, Z., Yang, Z., Liang, X., Salakhutdinov, R., and Xing, E. P. Toward controlled generation of text. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1587–1596. JMLR. org, 2017.

Jang, E., Gu, S., and Poole, B. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

Luong, M., Brevdo, E., and Zhao, R. Neural machine translation (seq2seq) tutorial. *https://github.com/tensorflow/nmt*, 2017.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pp. 311–318. Association for Computational Linguistics, 2002.

Ranzato, M., Chopra, S., Auli, M., and Zaremba, W. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*, 2015.

Sutskever, I., Vinyals, O., and Le, Q. V. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pp. 3104–3112, 2014.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems*, pp. 5998–6008, 2017.

Wang, W., Hu, Z., Yang, Z., Shi, H., and Xing, E. P. Differentiable expected BLEU for text generation, 2019. URL https://openreview.net/forum?id=S1x2aiRqFX.

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

Yang, Z., Hu, Z., Dyer, C., Xing, E. P., and Berg-Kirkpatrick, T. Unsupervised text style transfer using language models as discriminators. In *Advances in Neural Information Processing Systems*, pp. 7298–7309, 2018.

Yavuz, S., Chiu, C.-C., Nguyen, P., and Wu, Y. Calcs: Continuously approximating longest common subsequence for sequence level optimization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 3708–3718, 2018.

Zhukov, V., Golikov, E., and Kretov, M. Differentiable lower bound for expected bleu score. *arXiv preprint arXiv:1712.04708*, 2017.