On Text Style Transfer via Style Masked Language Models

Sharan Narasimhan, Pooja Shekar, Suvodip Dey, Maunendra Sankar Desarkar
Department of Computer Science and Engineering
Indian Institute of Technology Hyderabad, India

{sharan.n21, poojashekar15}@gmail.com
{cs19resch01003, maunendra}@iith.ac.in

Abstract

Text Style Transfer (TST) is performable through approaches such as latent space disentanglement, cycle-consistency losses, prototype editing etc. The prototype editing approach, which is known to be quite successful in TST, involves two key phases a) Masking of source style-associated tokens and b) Reconstruction of this source-style masked sentence conditioned with the target style. We follow a similar transduction method, in which we transpose the more difficult direct source to target TST task to a simpler Style-Masked Language Model (SMLM) Task, wherein, similar to BERT (Devlin et al., 2019a), the goal of our model is now to reconstruct the source sentence from its style-masked version. We arrive at the SMLM mechanism naturally by formulating prototype editing/transduction methods in a probabilistic framework, where TST resolves into estimating a hypothetical parallel dataset from a partially observed parallel dataset, wherein each domain is assumed to have a common latent style-masked prior. To generate this style-masked prior, we use “Explainable Attention” as our choice of attribution for a more precise style-masking step and also introduce a cost-effective and accurate “Attribution-Surplus” method of determining the position of masks from any arbitrary attribution model in O(1) time. We empirically show that this non-generative approach well suites the “content preserving” criteria for a task like TST, even for a complex style like Discourse Manipulation. Our model, the Style MLM, outperforms strong TST baselines and is on par with state-of-the-art TST models, which use complex architectures and orders of more parameters.

1 Introduction

Text Style Transfer (TST) can be thought of as a subset of the Controllable Language Generation Task (Hu et al., 2017) with tighter criteria. TST involves converting input text possessing some source style into an output possessing the target style, meanwhile preserving style-independent content and maintaining fluency. Style is usually defined by the class labels present in an annotated dataset, commonly considered to be Sentiment, Formality, Toxicity etc. We consider the unsupervised setting where only non-parallel datasets are available, i.e. the style-transferred output is not available. Past work focuses on various common paradigms such as disentanglement (Hu et al., 2017; Shen et al., 2017), cycle-consistency losses (Yi et al., 2020; Luo et al., 2019; Dai et al., 2019; Liu et al., 2021), induction (Narasimhan et al., 2022; Shen et al., 2020) etc. We focus on a sentence to sentence “transduction” (or prototype editing) method, a solution which naturally emerges when following a probabilistic formulation consisting of a single transduction model with a latent prior consisting of a style-absent corpus. Our approach is also inspired by BERT’s MLM (Devlin et al., 2019a), modified to incorporate style information to enable TST. This notion can be seen from various viewpoints.

Another MLM. By making use of various text attribution models, it is possible to determine the fraction of tokens that significant contribution to the style label. Masking out these tokens and training a bi-directional transformer encoder with self-attention with the task of reconstructing the original source style sentence is intuitive and closely resembles BERT-based MLM’s (Devlin et al., 2019a; Jin et al., 2020). Instead of random masking or perplexity based masking (Jin et al., 2020), by incorporating style-masking and concatenation of control codes that inform the encoder about the target style needed, the resultant MLM now resembles a rough TST model. This notion of “filling in only the style words” also does well to automatically ensure that

---

*A key difference to BERT, ours is not using a self-supervised approach of pre-training over large corpora and fine-tuning for a downstream task.*
Table 1: Examples of Sentiment and Discourse TST by the SMLM on the IMDb and SNLI datasets respectively.

<table>
<thead>
<tr>
<th>Task</th>
<th>Positive to Negative</th>
<th>Contradiction to Entailment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>this movie is by far one of the best urban crime dramas i've seen.</td>
<td>a woman is sitting outside at a table using a knife to cut into a sandwich. a woman is sitting inside.</td>
</tr>
<tr>
<td>Style Masked</td>
<td>this movie is by mask one of the mask urban crime mask i've seen.</td>
<td>a woman is sitting outside at a table using a knife to cut into a sandwich. a woman mask mask mask</td>
</tr>
<tr>
<td>Output</td>
<td>this movie is by far one of the worst urban crime garbage i've seen.</td>
<td>a woman is sitting outside at a table using a knife to cut into a sandwich. a woman is a outside</td>
</tr>
</tbody>
</table>

Contributions. a) We introduce the Style Masked Language Model (SMLM), another type of MLM capable of being of strong TST model by initially pretrained on an unsupervised same-style reconstruction task and fine-tuned for TST metrics. b) We use an accurate and effective style masking step and present an analysis to support our choice. c) We empirically show that the SMLM beats strong baselines and compares to state of the art, using far lesser parameters. d) We adopt a novel "Discourse" TST task by salvaging NLI datasets and show that our style-masking is also capable of working with more complex "Inter-sentence" styles to manipulate the flow of logic.

2 Related Work

Jin et al. (2021) and Hu et al. (2020) provide surveys detailing the current state of TST and lay down useful taxonomies to structure the field. In this section, we only discuss recent work similar to ours (a prototype editing approach) assuming the same unsupervised setting.

Li et al. (2018) present the earliest known work using the prototype editing method, in which a "delete" operation is performed on tokens based on simple count-based methods, and the retrieval of the target word is done by considering TF-IDF weighted word overlap. Malmi (2020) first train MLMs for the source and target domains and perform TST by first masking text spans where the models disagree (in terms of perplexity) the most, and use the target domain MLM to fill these spans. Wu et al. (2019b) introduce the Attribute-Conditional MLM, which most closely aligns to the working of the MLM, also uses an attention classifier for attribution scores, a count and frequency-based method to perform masking, and a pretrained BERT model fine-tuned on the TST task. Lee (2020) and Wu et al. (2020) also follows roughly the same pipeline but uses a generational transformer encoder-decoder approach and also fine-tunes using signals from a pretrained classifier. Wu et al. (2019a) uses a hierarchical reinforced sequence operation method is used to iteratively revise the words of original sentences. Madaan et al. (2020) uses n-gram TF-IDF based methods to identify style tokens and modify them as either "add" or "replace" TAG tokens, which are then substituted by the decoder to perform TST. Similar to the SMLM, (Xu et al., 2018) also uses attribution-based methods from a self-attention.
classifier. However, they use an LSTM (Hochreiter and Schmidhuber, 1997) based approach, one to generate sentences from each domain.

3 Method

3.1 Notation

Let $S$ denote the set of all style labels for a supervised dataset $D$ of the form $\{(x_0, l_0), (x_1, l_1) \ldots (x_n, l_n)\}$ where $x_i$ denotes the input sentence and $l_i \in S$ denotes the label corresponding to $x_i$. The set of all sentences of style $s$ in $D$ is denoted by $x^s = \{x_j : \forall j \text{ where}(x_j, s) \in D\}$. We use a special meta label $m_s$ to represent the "style-masked" label class having $s$ the original style. Subsequently, $x^{m_s}$ refers to the set of all style-masked sentences which had source style $s$. The set of all style-masked sentences from $D$ is given by $x^{m_s} = Union(x^{m_s} : \forall s \in S)$. Let $x^s$ denote the model’s estimation of $x^s$.

Probabilistic Overview. We treat the non-parallel dataset as a partially observed parallel dataset, where the SMLM has to estimate the entire joint distribution $P(x^s, x^s)$. The SMLM assumes that every sentence from each style domain is a result of, a) sampling from a latent style-masked prior, $p(x^{m_s})$, which we estimate using the Style-Masking model, $p(x^{m_s} | x^s, \theta_{SMLM})$ (conditional posterior), and is then b) conditionally reconstructed to form the sentence with target style using the SMLM, $q(s^*_i, x^*_i | x^{m_s})$ (the style-conditioned likelihood). The posterior in this case is easily estimable by using various intuitive methods. The likelihood function, represented by the SMLM, has a simple task of performing the same-style reconstruction task using a non-parallel dataset. Using control codes to guide which domain to sample from, enables the SMLM to also estimate the unseen section of the hypothetical parallel and therefore work as a rought TST model. The probabilistic formulation of the overall model is given in Fig. 1.

3.2 Explainable Attention as Attribution Markers

Many prototype editing methods use Vanilla Attention (VA) as attribute scores (Wu et al., 2019b; Zhang et al., 2018; Wu et al., 2020). However, it has been shown that attention is not explanation, at least for the purpose of human-aligned interpretability (Jain and Wallace, 2019). VA does not correlate well with other well-known attribution methods (such as Integrated Gradients Sundararajan et al. (2017)). We instead use "Explainable Attention" (EA) scores from a Diversity-LSTM classifier (Mohankumar et al., 2020; Nema et al., 2017) which have been shown to correlate better with other attribution methods. EA scores as attribution also correlates better with human judgement as compared to VA. We discuss the intuitive reasoning as to why VA over LSTM hidden states fail as attribution scores, the intuition of the Diversity-LSTM, its mechanism and loss equations in Section 8. We also quantitatively compare the efficacy of the style-masking step between EA and VA in Table 2.

3.3 The Style-Masking module

Even with having accurate attribution scores, effective style-masking requires careful selection of a policy which satisfies certain criteria. The primary criteria being that only tokens which significantly contribute to the style of a sentence must be masked, and leaving other tokens, to ensure content is also preserved. Similar to the masking policy in Wu et al. (2020), it is natural to consider a "top $k$ tokens" scheme in which the top $k$ tokens with highest attribution are masked. However, this static approach fails for sentences which do not have $k$ style-contributing tokens, leading to either partial style masking or erroneous masking of content tokens. For the same reason, even a sentence length aware scheme such as "top 15%" masking fails. Furthermore, all such policies require sorting, leading to O(nlogn) time complexity for style masking of each sentence in a batch.

An "Attention Surplus" policy. Let $A = \{A_1 \ldots A_n\}$ denote the attention distribution of a sentence of size $n$. Intuitively, we can reason that all "special" tokens which might contribute more to style should have an attribution greater than the average base attribution of the sentence, given by $A_{mean} = 1/n$. Generalising this further, We refer to tokens with $A_i \geq A_{baseline}$ as tokens with "attention surplus" w.r.t to a sentence-length sensitive baseline attention $A_{baseline}$ given by:

$$A_{baseline} = (1 + \lambda_c) * A_{mean}$$

where $\lambda_c$ is a hyperparameter of range $0 - 1.0$. This chosen threshold $A_{baseline}$ is sensitive to the sentence length as well and subsequently ensures that the number of style-significant tokens can be dynamically determined, without need of an elaborate algorithm. As a sanity check, we observe
that even in the adversarial case where all tokens might be equally important to style, $A$ resolves into a UniformDistribution$(n)$ and our policy correctly resorts to masking all tokens*. Let $Mask$ denote the token mask matrix of size $n$ initialised with zeros.

$$Mask[A_i \geq A_{\text{baseline}}] = 1 \quad (2)$$

Using a vectorised approach as given in the above equation, we can achieve style-masking for an entire batch in $O(1)$ as compared to $O(n\log n)$ per sentence in previous approaches.

### 3.4 A Style-aware MLM via Transformer Encoder Block

**The SMLM.** For the sake of convenience in notation, we assume binary style labels, $S = \{0, 1\}$

The task of TST now resolves into estimating $q(x^s|x^{ms}, \theta_{SMLM})$. The style-masking module from the previous step gives us access to the discrete prior $p(x^{ms})$ i.e. available as pairs of the form $(x^s_i, x^{ms}_i)$. The task of reconstructing a sentence $x_i^{m0}$ e.g. "The food was <blank>" into the original sentence $x_i^0$, e.g. "The food was good" resembles a Masked Language Model (MLM) objective similar to Devlin et al. (2019b) where additionally the target style is contextualised as well. We thus refer to this task of "Style-aware" MLM (SMLM), which consist of estimating two distributions i.e. $q(x^s|x^{ms}, \theta_{SMLM})$ and $q(x^s|x^{ms}, \theta_{SMLM})$.

**Control Codes.** We concatenate two special metatokens $[dst]$, $[src]$ to $x_i^{ms}$ i.e. the source and destination style labels respectively. This is denoted by the function $\text{concat}(x_i^{ms}, [src], [dst])$. Choice of $[src]$, $[dst]$ triggers the SMLM model to sample different distributions, either $q(x^s|x^{ms})$ or $q(x^s|x^{ms})$, corresponding to the "Same-Style" Reconstruction and TST Task respectively. The final model is defined as:

$$q_{SMLM} = \begin{cases} q(x_i^s|x_i^{ms}, \theta_{SMLM}), & \text{if } [src] = [dst] \\ q(x_i^s|x_i^{ms}, \theta_{SMLM}), & \text{if } [src] = [src] \end{cases}$$

**2-Stage Training.** Similar to Liu et al. (2021), we use a two stage method of training our TST model. In the first stage, we bootstrap a rough TST model and then finetune to improve selected metrics for the TST task.

\*Assuming $\lambda_s = 0$, whereas in practice we find $\lambda_s = 0.15, 0.5$ giving optimal masking for the Sentiment and Discourse TST, respectively. More is discussed in 5.2

\*1In theory, this can be extended to any number of styles.

---

**Figure 1:** Probabilistic overview of Transduction Model assuming common Latent Style-Masked Variable as prior.

### 3.4.1 Bootstrapping on Same Style Reconstruction Task

The SMLM model is first trained to estimate the likelihood $q(x^s|x^{ms})$ and therefore set $[src] = [dst] = s$ in $\text{concat}$. This corresponds to the task of reconstructing original sentences $x^s$ from their corresponding style-masked sentence $x^{ms}$ produced by the SM module. Due to the nature of the SMLM task in which input and output pairs $(x^{ms}, x^s)$ always have the same length, the Transformer (Vaswani et al., 2017) encoder block possesses as a natural choice since it enforces same-dimension output transformations\(^\dagger\). The Transformer Encoder Block $E$ minimizes the NLL given below:

$$L_{BS}(\theta_{SMLM}) = -\log q_{SMLM}(x^s|x^s)$$

### 3.4.2 Fine-tuning for TST

We now fine-tune the SMLM’s performance for the certains metric in the TST task i.e. $q(x^s|x^{ms})$ which is triggered by setting $[dst] = [src]$. We fine-tune for TST\% by jointly training a style classifier $cls$ by minimising the NLL taking the same-style reconstructions of SMLM i.e. $x^s$ as input.\(^\ddagger\)

$$L_{cls}(\theta_{cls}) = -\log p_{cls}(s|x^s)$$

\(^\dagger\)This however constrains the output size to always match the input. Consequently, our SMLM model is not generational. In section 6.6, we further motivate this decision.

\(^\ddagger\)We use a single feed-forward layer with input being the average of the last layer embeddings of $X^t$ (excluding the meta-labels).
This classifier now provides a supervision signal that the SMLM uses to align \( x^s \) to style attribute \( s \). Whereas the discriminator tries to discourage this. This is formulated as \( \min_{\theta_{SMLM}} \max_{\theta_{cls}} -\log p_{cls}(s|x^s) \).

We do not fine-tune for content-preservation or fluency as we do not observe any lacking in these metrics after bootstrapping.

4 Datasets and Tasks

We report the split and label wise statistics of each dataset in 14.

**Sentiment TST:** Following many past studies, we evaluate our model for the sentiment TST task using three review datasets, Yelp, Amazon and IMDb. All three datasets are annotated with two labels corresponding to positive or negative reviews and are non-parallel.

**Discourse TST:** To illustrate that our model also performs well in such situations, we introduce the Discourse TST task by performing TST on the SNLI (Bowman et al., 2015) dataset. Each instance consists of two sentences, which either contradict, entail (or agree) or are neutral (no relationship) to each other. We consider the task of converting the discourse style between two sentences, from contradiction to entailment and vice-versa. Some TST tasks are more complex than others and have different levels of granularity (Lyu et al., 2021). Unlike the sentiment task, which is "intra-sentence", where the style can be attributed to a select set of words, the discourse task is "inter-sentence" and requires the model to be cognizant of the context and detect the flow of logic rather than just a few style words (especially for the Contradiction to Entailment Task).

5 Analysis of Style-Masking approach

In this section, we evaluate and justify our choice of style-masking architecture, i.e. "Explainable Attention" + "Attention-Surplus" masking policy. We explore the question "What actually constitutes a good style-masking step?". Intuitively, we can reason that our style-masking approach must a) produce accurate attribution scores for each token b) use an appropriate masking policy that determines which tokens to mask using these attribution scores. The final style-masked sequences (latent prior to the SMLM) must actually be devoid of style information and accurate, i.e. not done at the expense of content information.

5.1 Quant. Analysis of Various Attribution Methods

We consider various other attribution methods for our analysis i.e. Vanilla Gradients, Integrated Gradients (Sundararajan et al., 2017), Vanilla Attention, Attention * X (or inputs) and Explainable Attention (Mohankumar et al., 2020). We do not consider techniques such as LIME (Ribeiro et al., 2016), LRP (Bach et al., 2015), DeepLIFT (Shrikumar et al., 2017) as they are relatively more computationally expensive during inference time. As the style-masking policy, use the "attribution-surplus" policy to determine the final style-masked sequences.

In **Table 2**, we compute the Accuracy% and s-BLEU on the final style-masked sequences produced by each attribution method on the test split for all four datasets. We can reason that an ideal style-masking method should be able to produce sentences that completely mask out style, thereby fooling a pretrained classifier (minimizing its Accuracy%) and also preserving content information (maximizing the s-BLEU between the source and style-masked sentences). We see that though Vanilla Attention is able to generally produce the lowest Accuracy%, it does so at the expense of preserving content, reflected as lower s-BLEU compared to EA. EA, on the other hand, has the best content-preserving style masking throughout all datasets and comes as a close second in terms of
5.2 Effect of $\lambda_\epsilon$ on Style-Masking

We can intuitively reason that s-BLEU metric of the style-masked sentences serves as a rough upper bound of the s-BLEU we achieve on the output sentences after TST. As expected (from Eq. 1), we observe a positive correlation between $\lambda_\epsilon$ and both s-BLEU and Accuracy% as seen in Table 3. It is desirable to carefully choose $\lambda_\epsilon$ to be a high enough to boost future s-BLEU scores on the TST task and also ensure that the sentences are sufficiently style masked with low Accuracy% scores. Only manual checking, we observed that $\lambda_\epsilon = 0.15$ served best to accurately style-mask sentences for the Yelp, IMDb and Amazon datasets. SNLI required a higher $\lambda_\epsilon$ of 0.5 to ensure content information was preserved appropriately.

6 Experiments

We perform TST on all the datasets delineated in Section 14 and analyse the results.

6.1 Automatic Evaluation Metrics

Typically, metrics for TST include Style Transfer Accuracy (using a pretrained classifier), BLEU for content preservation, and perplexity (using a pretrained LM) to check the fluency of outputs. Xu et al. (2020) show that this traditional set of metrics can be gamed. For fluency, we use the "Naturalness" metric (Mir et al., 2019) instead of PPL as it is shown to correlate better with human judgement. Apart from using BLEU for content preservation, we also report METEOR (Banerjee and Lavie, 2005), ROUGE-L (Lin, 2004) and CIDEr (Vedantam et al., 2015) scores in Table 11.

6.2 Baselines and Hyperparameters

Baseline Selection. As baselines, we choose DirR (Liu et al., 2021), Stable (Lee, 2020), Transforming (Sudhakar et al., 2019), Tag (Madaan et al., 2020), CrossAligned (Shen et al., 2017), CycleRL (Xu et al., 2018), StyleEmbedding (Fu et al., 2018), D&R (Li et al., 2018) and CycleMulti (Dai et al., 2019). For the hyperparameters of each baseline, we consider the optimal parameters of the best models for each dataset reported in each respective work. Whenever available, we directly make use of the flagship TST outputs published as part of the original work of each reference paper to ensure that a fair comparison is done.

SMLM Versions. We consider three version of the SMLM model i.e. without fine-tuning (SMLM), with fine-tuning for 1 epoch (SMLM-F). We also consider a Generational version of SMLM with an additional auto-regressive Transformer-based decoder (SMLM-G).

SMLM Hyperparameters. SMLM consists of a lightweight Transformer Encoder block with 2 layers, 8 heads and embeddings of size 512. We find that training it for 15 epochs on the Same-Style Reconstruction Task and fine-tuning it for 1 epoch on for TST produces optimal results. During fine-tuning, $\lambda_{sta}$ was set to 1 and gradient-clipping with a threshold of $10^{-3}$ was set to prevent gradients from exploding.

Diversity-LSTM Hyperparameters. As mentioned earlier, we use the "explainable" attention scores of the Diversity-LSTM classifier in our style-masking step. The Diversity-LSTM’s mechanism, auxiliary losses and hyperparameters chosen is presented in Section 8.

Setup and #Parameters. The Tag (Madaan et al., 2020) and DirR (Liu et al., 2021) models (the two best performing baselines) have 50M and 1.5B parameters respectively. The SMLM has 45M parameters, 30x lesser parameters than DirR’s fine-tuned GPT-2 model, and roughly the same number of parameters as Tag, but outperforming it in the IMDb and SNLI datasets. We report details on training time required and infrastructure used in Section 8.
Table 2: Comparison of quality of Style-Masking produced using various attribution models. We found that \( \lambda = 0 \) worked best with all gradient-based methods. For attention based methods (VA and EA), we found that \( \lambda = 0.15, 0.5 \) worked best for {Yelp, IMDb, Amazon}, SNLI respectively.

### 6.3 Quant. metrics for TST task

We compute TST\% (percentage of sentences with target style) using a Bi-LSTM based pretrained classifier trained on each dataset (refer 8 for classifier details). r-BLEU and s-BLEU refer to the BLEU score taken between the output sentences and the human reference and ground truth sentences, respectively. For fluency, we measure the mean "Naturalness" score (the "Nat." column) as the mean classification score of a pretrained fluency discriminator\(^1\). We also add a "Mean" score consisting of the average of TST\%, s-BLEU and Nat. (normalised to 100) columns to denote a rough measure of the overall quality of each TST model.

### 6.4 Sentiment TST

Sentiment TST is performed on Yelp (Table 3), IMDb (Table 5) and Amazon (Table 4). We observe that for Yelp and IMDb, SMLM-FT and DirR are the best performing models according to the Mean score. In IMDb, DirR performs better than SMLM-FT in content preservation metrics but slightly lags behind in naturalness scores. In Amazon, SMLM-FT achieves a significantly high TST\% score at a reasonable s-BLEU of 60.9. In Amazon, Tag and SMLM-FT are the best performing, with Tag having a significantly high TST\% score at competitive content and fluency metrics. Overall we observe SMLM-FT, DirR and Tag as the best performing models.

### 6.5 Discourse TST

It might be presumed that Prototype editing methods such as SMLM are only capable of working well on "course-grained" styles, i.e. where the presence of style is determined by the presence of a set of certain words that convey the style. To inspect if this is true and gauge the ability of the SMLM to operate on more cognitive and complex tasks, we consider "Discourse TST" by using Natural Language Inference (NLI) datasets.

We report statistics for the SMLM in Table 6. We observe that the SMLM-FT model does well overall in this task and obtains a strong mean score of 85.53 (higher than the sentiment TST tasks). It lags behind a little mainly in the TST\% metric, but with a strong s-BLEU and Naturalness score. We also report qualitative examples of the discourse TST task in Table 9 and 12.

### 6.6 Ablation Studies

**Without Finetuning.** As expected, The "SMLM" model in Table 3, 5, 4 and 6 has substantially better r-BLEU and s-BLEU scores compared to the fine-tuned SMLM but lacks in TST\% metrics.
<table>
<thead>
<tr>
<th>Model</th>
<th>TST%</th>
<th>s-BLEU</th>
<th>Nat. Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>DirR</td>
<td>80.3</td>
<td>67.9</td>
<td>0.92</td>
</tr>
<tr>
<td>cycleMulti</td>
<td>67.2</td>
<td>73.7</td>
<td>0.93</td>
</tr>
<tr>
<td>SMLM</td>
<td>66.8</td>
<td>69.2</td>
<td>0.92</td>
</tr>
<tr>
<td>SMLM-G</td>
<td>68.9</td>
<td>65.6</td>
<td>0.93</td>
</tr>
<tr>
<td>SMLM-FT</td>
<td>87.9</td>
<td>60.9</td>
<td>0.92</td>
</tr>
</tbody>
</table>

Table 5: Quantitative metrics for the IMDb dataset.

<table>
<thead>
<tr>
<th>Model</th>
<th>TST%</th>
<th>s-BLEU</th>
<th>Nat. Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tag</td>
<td>48.3</td>
<td>90.2</td>
<td>0.98</td>
</tr>
<tr>
<td>SMLM</td>
<td>52.2</td>
<td>88.5</td>
<td>0.98</td>
</tr>
<tr>
<td>SMLM-G</td>
<td>58</td>
<td>86.7</td>
<td>0.98</td>
</tr>
<tr>
<td>SMLM-FT</td>
<td>76.3</td>
<td>86.3</td>
<td>0.94</td>
</tr>
</tbody>
</table>

Table 6: Quantitative metrics for the SNLI dataset.

Using Decoder. The generational non fine-tuned "SMLM-G" model performs quite similarly to the non-fine-tuned decoder-less "SMLM" model with respect to the mean score. Practically, we observed that trying to fine-tune SMLM-G for a better TST% metric is difficult and leads to gradient exploding even with a large amount of gradient clipping. The SMLM-FT model appears to be more stable during fine-tuning with a lesser computational expense.

6.7 Additional Content Metrics.

Past work does not tend to clarify the meaning and prioritise the presence of "content-preservation" abilities in TST models Lee et al. (2021). In this effort, a more thorough analysis of content preservation abilities of DirR, Tag and SMLM-FT is given in Section 11.

6.8 Qualitative Examples

Examples of the TST task performed using the SMLM for the IMDb and SNLI dataset are given in Table 8 and 9 respectively.

6.9 Human Evaluations

We only consider the relatively unexplored Discourse TST (Entailment to Contradiction and vice versa) task for human evaluations. We were unable to reproduce the DirR baseline to run over the SNLI dataset. Therefore, we only compare the next strongest performing models i.e., Tag and SMLM. Three volunteers were given the task of voting on 200 instances (equally split for the E to C and C to E task) from the test set. A vote consists of four options i.e., "Model 1 better", "Model 2 better" or "Both Good", "Both Bad", where the models were randomised. To determine the decision an each instance, a majority considered by taking three separate votes, one from each volunteer. In the case of no majority, "No agreement" decision was taken.

<table>
<thead>
<tr>
<th>Direction</th>
<th>Tag</th>
<th>SMLM</th>
<th>Both Good</th>
<th>Both Bad</th>
<th>No Agreement</th>
</tr>
</thead>
<tbody>
<tr>
<td>E to C</td>
<td>11</td>
<td>55</td>
<td>2</td>
<td>18</td>
<td>8</td>
</tr>
<tr>
<td>C to E</td>
<td>8</td>
<td>44</td>
<td>35</td>
<td>11</td>
<td>11</td>
</tr>
</tbody>
</table>

Table 7: Human Evaluations done to compare Tag and SMLM on Discourse TST task on SNLI dataset. “E” and “C” denote “Entailment” and “Contradiction” respectively.

As seen in Table 7, the SMLM performs better in both tasks by a significant margin. We report some qualitative examples taken from the evaluation in Table 12.

7 Conclusion

We introduce the SMLM, a modification of the standard MLM, which we show is capable of performing TST by using a style-masked input and performing a simple same-style reconstruction task with a lightweight Transformer Encoder block. On fine-tuning the SMLM for the TST%, it is on par with state of the art models with orders of more parameters and sophisticated architectures in the Sentiment TST task. We show that complex styles such as flow of logic/ discourse can be manipulated even with using this simple style masking assumption. We empirically show that the SMLM performs well in this Discourses Manipulation task and outperforms another strong baseline in this task, also seen through human evaluations.

8 Limitations

The apparent limitation with all prototype editing models, including the SMLM, is that it encourages the model to only fill in necessary style words and preserve the length and structure of the original sentence. In the case of SMLM, the word-to-word input-output mapping while training the encoder prevents the output sentence length from changing. Though it can be argued that this even works for a relatively cognitive style like discourse, in the future, there might exist styles which explicitly require the addition/deletion of words/phrases in order to successfully alter the style. Future work will therefore focus on enable variable length TST outputs, similar to the (Madaan et al., 2020) approach or by incorporating a padded masked language model (Malmi, 2020).
References


Ning Dai, Jianze Liang, Xipeng Qiu, and Xuanjing Huang. 2019. Style transformer: Unpaired text style transfer without disentangled latent representation. In *ACL*.


Sarthak Jain and Byron C. Wallace. 2019. Attention is not explanation. In *NAACL*.


Appendix

Qualitative examples of TST

Details of Pretrained Classifier

We use a Bi-LSTM as our choice of classifier as it performs comparably to FastText (Joulin et al., 2017) and outperforms it in the SNLI dataset. A comparison of the two models is given in Table 8.

Additional Content Preservation Metrics

We present more content preservation metrics in Table 8 to compare the top three performing models i.e., SMLM, Tag (Madaan et al., 2020) and DirR (Liu et al., 2021).

Additional examples of Discourse TST

We focus more again on the Discourse TST task and report more examples of the SMLM and the Tag (Madaan et al., 2020) baseline in Table 12.

Computational Expense and Infrastructure used

The most parameter-heavy SMLM model was from the SNLI dataset. Therefore we report statistics for this model to gauge the overall computational expenses the SMLM demands. The model has 45 million parameters and each epoch took approximately 224 seconds to train on an Nvidia V100-SMX2 GPU and an Intel(R) Xeon(R) E5-2698 CPU. For complete details, we will make the code open source which will also contain the models we trained along with log files with all metadata about the model architecture and training.

Ethics Statement

Any TST model can be used for illicit purposes. Therefore, it is important we keep in mind a code of ethics (e.g. https://www.acm.org/code-of-ethics). We will make all our code
ben affleck is back to making the same truly great acting films. this movie is by far one of the worst urban crime garbage i’ve seen.

Table 8: Example of Sentiment TST on the IMDb dataset.

<table>
<thead>
<tr>
<th>Direction</th>
<th>Negative to Positive</th>
<th>Positive to Negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>ben affleck is back to making the same boring bad acting films.</td>
<td>this movie is by far one of the best urban crime dramas i’ve seen.</td>
</tr>
<tr>
<td>Style</td>
<td>ben affleck is back to making the same &lt;mask&gt; &lt;mask&gt; acting films.</td>
<td>this movie is by &lt;mask&gt; one of the &lt;mask&gt; urban crime &lt;mask&gt; i’ve seen.</td>
</tr>
<tr>
<td>Output</td>
<td>ben affleck is back to making the same truly great acting films.</td>
<td>this movie is by far one of the worst urban crime garbage i’ve seen.</td>
</tr>
</tbody>
</table>

Table 9: Example of Discourse TST on the SNLI dataset.

Table 10: Comparison of FastText, Bi-LSTM models for classification task on all datasets.

Table 10: Comparison of FastText, Bi-LSTM models for classification task on all datasets.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Model</th>
<th>Acc.%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yelp</td>
<td>FastText</td>
<td>97.6</td>
</tr>
<tr>
<td></td>
<td>Bi-LSTM</td>
<td>97</td>
</tr>
<tr>
<td>IMDb</td>
<td>FastText</td>
<td>99.35</td>
</tr>
<tr>
<td></td>
<td>Bi-LSTM</td>
<td>99</td>
</tr>
<tr>
<td>Amazon</td>
<td>FastText</td>
<td>92.1</td>
</tr>
<tr>
<td></td>
<td>Bi-LSTM</td>
<td>93</td>
</tr>
<tr>
<td>SNLI</td>
<td>FastText</td>
<td>72.5</td>
</tr>
<tr>
<td></td>
<td>Bi-LSTM</td>
<td>84</td>
</tr>
</tbody>
</table>

The Diversity-LSTM and Explainable Attention

Effective style-masking requires an attribution model with a high degree of plausibility, which motivates our use of "explainable" attention scores Mohankumar et al. (2020) as choice for the style-masking step.

Why not use standard attention? Vanilla attention scores do not serve as accurate attribution scores. Attention scores over RNN hidden states for the classification task do not correlate well with other standard interpretation metrics (Jain and Wallace, 2019), such as gradient and occlusion based methods. Feeding alternative adversarial/random attention distributions lead to only a modest effects are the model’s decision (Wiegrefe and Pinter, 2019). However Wiegrefe and Pinter (2019) shows that these adversarial distributions, if properly produced, do induce poorer performance showing that vanilla attention is still partially faithful to its explanation. Mohankumar et al. (2020) postulate that attention scores over hidden states (H) are not explainable due to information mixing and subsequent entanglement/coupling and mutual information among H in RNNs. To mitigate this entanglement, diversity driven learning (inspired by results in Nema et al. (2017)) is enforced among H. This promotes the attention mechanism over such diversity-enforced H to satisfy "faithfulness" and "plausibility" properties when interpreted as attribution scores, which we refer to as "Explainable attention" (EA). Mohankumar et al. (2020) empirically show that EA does not suffer any loss in performance in the downstream task. Supporting plausibility, a) EA scores correlate better with strong attribution tools such as Integrated Gradients b) On analysis over POS tags, EA attends more to tags which are contextually important w.r.t the given task and c) Correlates better to human judgement than vanilla attention.

The Diversity Driven LSTM. The Diversity LSTM consists of an LSTM-based classifier with attention (Bahdanau et al., 2015) over the H. The final context vector is fed through a feedforward layer to generate the output.

$$\tilde{\alpha}_t = v^T \tanh(W_h h_t + b) \quad \forall t \in [1, m]$$

$$\alpha_t = \text{softmax}(\tilde{\alpha}_t)$$

$$e_t = \sum_{t=1}^{m} \alpha_t h_t$$

To enforce the H of the LSTM to be "diverse" i.e. more disentangled w.r.t each other, the conicity
### Table 11: Content Preservation metrics for all datasets comparing top performing models

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Model</th>
<th>METEOR</th>
<th>ROUGE-L</th>
<th>CIDEr</th>
<th>Embedding Avg. Cosine Sim.</th>
<th>Vector Extrema Cosine Sim.</th>
<th>Greedy Matching Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yelp</td>
<td>SMLM-FT</td>
<td>0.376</td>
<td>0.739</td>
<td>4.934</td>
<td>0.939</td>
<td>0.767</td>
<td>0.867</td>
</tr>
<tr>
<td></td>
<td>DirR</td>
<td>0.444</td>
<td>0.83</td>
<td>5.813</td>
<td>0.969</td>
<td>0.867</td>
<td>0.926</td>
</tr>
<tr>
<td></td>
<td>Tag</td>
<td>0.362</td>
<td>0.707</td>
<td>4.326</td>
<td>0.934</td>
<td>0.765</td>
<td>0.867</td>
</tr>
<tr>
<td>IMDb</td>
<td>SMLM-FT</td>
<td>0.414</td>
<td>0.8</td>
<td>5.657</td>
<td>0.964</td>
<td>0.782</td>
<td>0.921</td>
</tr>
<tr>
<td></td>
<td>DirR</td>
<td>0.469</td>
<td>0.823</td>
<td>6.612</td>
<td>0.976</td>
<td>0.821</td>
<td>0.929</td>
</tr>
<tr>
<td></td>
<td>Tag</td>
<td>0.453</td>
<td>0.835</td>
<td>6.548</td>
<td>0.966</td>
<td>0.781</td>
<td>0.917</td>
</tr>
<tr>
<td>Amazon</td>
<td>SMLM-FT</td>
<td>0.563</td>
<td>0.906</td>
<td>8.297</td>
<td>0.986</td>
<td>0.886</td>
<td>0.96</td>
</tr>
<tr>
<td></td>
<td>DirR</td>
<td>0.606</td>
<td>0.944</td>
<td>8.619</td>
<td>0.992</td>
<td>0.921</td>
<td>0.972</td>
</tr>
<tr>
<td></td>
<td>Tag</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SNLI</td>
<td>SMLM-FT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>DirR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Tag</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 12: Examples of Discourse TST on SNLI of SMLM vs Tag (Madaan et al., 2020)

<table>
<thead>
<tr>
<th>Direction</th>
<th>Entailment to Contradiction</th>
<th>Contradiction to Entailment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>a black women holding a sign that says free hugs in the city . a woman is holding a sign</td>
<td>a tan dog chases a black and white soccer ball . a dog is chasing after a cat</td>
</tr>
<tr>
<td>Output (Tag)</td>
<td>a black women holding a sign that says free hugs in the city . a woman is holding a sign</td>
<td>a tan dog chases a black and white soccer ball . a dog is chasing after a sport</td>
</tr>
<tr>
<td>Output (SMLM-FT)</td>
<td>a black women holding a sign that says free hugs in the city . a woman is holding a book</td>
<td>a tan dog chases a black and white soccer ball . a dog is outside after a ball</td>
</tr>
</tbody>
</table>

### Diversity-LSTM Hyperparameter Selection

Table 8 gives details of the Diversity-LSTM classifier used during the style-masking step. We also note that the Diversity-LSTM’s performance is comparable to a Bi-LSTM and FastText classifier (as shown in 8).

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Acc. %</th>
<th>λ_{con}</th>
<th>λ_{con}</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yelp</td>
<td>96</td>
<td>0.06</td>
<td>10</td>
</tr>
<tr>
<td>IMDb</td>
<td>100</td>
<td>0.09</td>
<td>10</td>
</tr>
<tr>
<td>Amazon</td>
<td>89</td>
<td>0.03</td>
<td>20</td>
</tr>
<tr>
<td>SNLI</td>
<td>82</td>
<td>0.18</td>
<td>10</td>
</tr>
</tbody>
</table>

### Table 13: Classification task statistics and choice of λ_{con} for each dataset.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Style Label</th>
<th>Train</th>
<th>Dev</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yelp</td>
<td>Positive</td>
<td>266K</td>
<td>2000</td>
<td>500</td>
</tr>
<tr>
<td></td>
<td>Negative</td>
<td>177K</td>
<td>2000</td>
<td>500</td>
</tr>
<tr>
<td>IMDb</td>
<td>Positive</td>
<td>178K</td>
<td>2000</td>
<td>1000</td>
</tr>
<tr>
<td></td>
<td>Negative</td>
<td>187K</td>
<td>2000</td>
<td>1000</td>
</tr>
<tr>
<td>Amazon</td>
<td>Positive</td>
<td>277K</td>
<td>985</td>
<td>500</td>
</tr>
<tr>
<td></td>
<td>Negative</td>
<td>179K</td>
<td>1015</td>
<td>500</td>
</tr>
<tr>
<td>SNLI</td>
<td>Entailment</td>
<td>183K</td>
<td>3329</td>
<td>3368</td>
</tr>
<tr>
<td></td>
<td>Contradiction</td>
<td>183K</td>
<td>3278</td>
<td>3237</td>
</tr>
</tbody>
</table>

Table 14: Split and label wise statistics of each dataset.
Training Algorithm

1 foreach bootstrap_epoch do
   2   foreach mini-batch do
   3       minimize $L_{BS}$ w.r.t $\theta_{SMLM}$
   4   end
   5 end

6 foreach finetune_epoch do
   7   foreach mini-batch do
   8       minimize $L_{BS}$ w.r.t $\theta_{SMLM}, \theta_{cls}$
   9       minimize $L_{cls}$ w.r.t $\theta_{cls}$
  10   end
  11 end

**Algorithm 1**: Training process.