本文汇总几篇 text style transfer 这个任务在一些非常规场景下的模型：

1． Few shot / domain adaptive text style transfer

emnlp 19 的这篇 Domain Adaptive Text Style Transfer 希望可以用额外的数据集（source domain）来增强在目标数据集（target domain）上的表现。这篇文章所基于的基础模型仍然是基于对抗的文本风格迁移模型，包含两部分，一部分训练 AE 想要重建输入文本，一部分对于风格迁移后的文本，希望能够骗过分类器。

$$p_D(\widetilde{x}_i|c_i,\widetilde{l}_i)=\prod_{t=1}^{T}p_D(\widetilde{x}_i^t|\widetilde{x}_i^{<t},c_i,\widetilde{l}_i)$$

$$L_{ae}^{\mathcal{T}}=-\mathop{\mathbb{E}}_{x_i\sim\mathcal{T}}\log p_D(x_i|c_i,l_i)$$

$$L_{style}^{\mathcal{T}}=-\mathop{\mathbb{E}}_{\widetilde{x}_i\sim p_D(\widetilde{x}_i|c_i,\widetilde{l}_i)}\log P_{C^{\mathcal{T}}}(\widetilde{l}_i|\widetilde{x}_i)$$

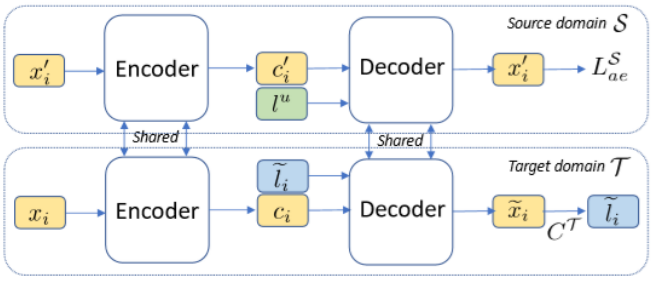
以上所示是在有 label 的目标数据集上（如 yelp）进行 transfer 训练的 loss。而如果我们还有其他数据集，则可以用来增强这个模型。这可以分为两种情况，第一种是 source domain 没有 style 的 label，仅仅是一个大数据集来增强 AE 部分的训练；第二种情况是 source domain 有 label，而且和 target domain 的 label 一致。

在第一种情况下，由于源数据集上没有 style 的 label，而训练 AE 的时候会根据不同的 style 训练不用的 decoder，或者将 style label/emb 也作为 decoder 的输入。为了在这种情况下训练 AE，本文的设计是增加一个 unknown style label，以情感转换为例，这时的 style 分为 pos、neg、unk。此时训练目标为：

$$L_{ae}^{\mathcal{S}}=-\mathop{\mathbb{E}}_{x_i'\sim\mathcal{S}}\log p_D(x_i'|c_i',l^u)$$

$$L_{\text{DAST-C}}=L_{ae}^{\mathcal{T}}+L_{style}^{\mathcal{T}}+L_{ae}^{\mathcal{S}}$$

值得注意的是，这个模型是可以应用在 few shot 的场景下的，few shot 是指只有很少的有风格的样本，比如正负各自 2k，这样的数据量是无法训练出一个好的生成模型的，但是我们可以用大量的没有 style label 的样本来进行增强。虽然该模型适用于这两种场景，但是两个场景目的是不同的，前者是希望比仅有 yelp 数据集时有更好的 transfer 效果，后者是希望在有 style 的数据不足的情况下也能学到迁移模型。

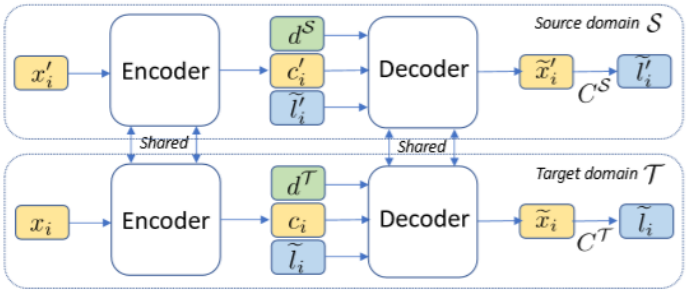


第二种情况，一个简单的方法就是将两个数据集直接混在一起，用同一个模型训练。但是由于不同的 domain 之间是有区别的，这样做无法进行 domain specific transfer，比方说同时使用 IMDB movie reviews 和 yelp restaurant reviews 训练，可能会产生这样的生成结果：the pizza is dramatic。而本文采取的方法是将一个可被训练的 domain vector 作为 decoder 的输入：

$$L_{ae}^{\mathcal{S},\mathcal{T}} = - \mathop{\mathbb{E}}_{x'_i \sim \mathcal{S}} \log p_D(x'_i | c'_i, d^{\mathcal{S}}, l'_i) - \mathop{\mathbb{E}}_{x_i \sim \mathcal{T}} \log p_D(x_i | c_i, d^{\mathcal{T}}, l_i),$$

注意到两个 domain 的 style 空间是相同的，比方说都是包含情绪的正或者负，但是本文分别在每个 domain 训练了一个分类器:

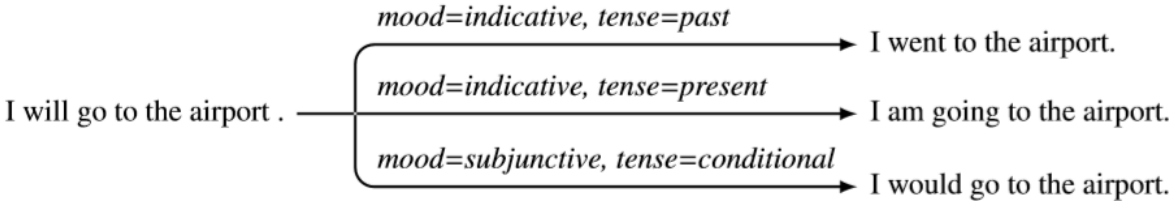
$$L_{style}^{\mathcal{S},\mathcal{T}} = - \mathop{\mathbb{E}}_{\widetilde{x}'_i \sim p_D(\widetilde{x}'_i | c'_i, d^{\mathcal{S}}, \widetilde{l}'_i)} \log P_{C^{\mathcal{S}}}(\widetilde{l}'_i | \widetilde{x}'_i) - \mathop{\mathbb{E}}_{\widetilde{x}_i \sim p_D(\widetilde{x}_i | c_i, d^{\mathcal{T}}, \widetilde{l}_i)} \log P_{C^{\mathcal{T}}}(\widetilde{l}_i | \widetilde{x}_i)$$



而我们还可以想到第三种场景，也就是 source domain 和 target domain 的风格不相同时，能否在 source data 上进行 target 风格的迁移，也就是 out-of-domain 的情况。

2. multiple attribute text rewriting

Nips2018 的这一篇 Content preserving text generation with attribute controls 给出了一种可以在多种属性上同时进行 transfer 的方法，对于一个数据集，如果同时有多种属性的 label，比如同时有情绪、时态、性别等，则可以训练一个模型，同时控制某生成文本的多种属性:



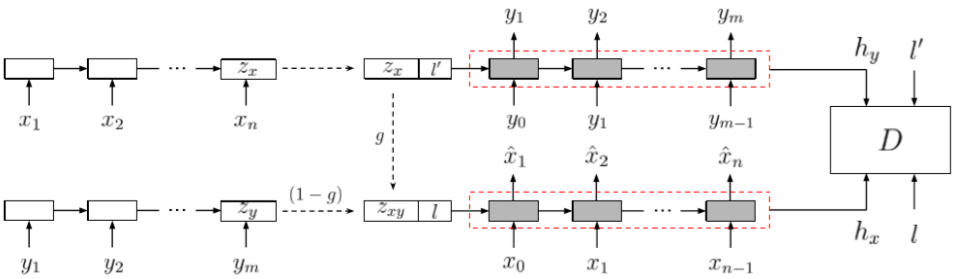
本文将多个属性的 label 处理成多个 one-hot 向量，然后拼接起来，称为 attribute vertor。本工作基本上还是一个基于对抗方法的模型，为了保持 transfer 之后的内容一致，用了以下三个 loss:

$$\mathcal{L}^{ae}(x, l) = -\log p_G(x | z_x, l)$$

$$\mathcal{L}^{bt}(x, l) = -\log p_G(x | z_y, l)$$

$$\mathcal{L}^{int} = \mathbb{E}_{(x,l) \sim p_{\text{data}}, y \sim p_G(\cdot | z_x, l')}[-\log p_G(x | z_{xy}, l)]$$

其中第一个 ae loss，第二个是 back-translation loss，第三个是为了避免第二个 loss 难以训练，将两者得到的隐变量进行插值，用来重建文本。



为了使得风格可以成功迁移，使用了如下对抗 loss：

$$\mathcal{L}^{\text{adv}} = \min_G \max_D \mathbb{E}_{\substack{(x,l)\sim p_{\text{data}} \\ y\sim p_G(\cdot|z_x,l')}} [2\log D(h_x,l) + \log(1-D(h_y,l')) + \log(1-D(h_x,l'))]]$$

可以看到，生成的句子不真实或者生成的句子与风格不一致都作为负例。