Computer Science and Application 计算机科学与应用, 2020, 10(10), 1888-1899
Published Online October 2020 in Hans. http://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2020.1010199

Hans汉斯

基于神经网络的文本风格转换

郝志峰[1,2]，陈渝升[1*]，蔡瑞初[1]，温　雯[1]，王丽娟[1]

[1]广东工业大学计算机学院，广东 广州
[2]佛山科学技术学院，数学与大数据学院，广东 佛山
Email: *zfhao@gdut.edu.com

收稿日期：2020年10月8日；录用日期：2020年10月23日；发布日期：2020年10月30日

摘　要

文本风格转换在书面创作、品牌推广等许多方面具有良好的应用前景，近年来也逐渐成为研究热点。现有的文本转换工作对风格表示简单，无法适应文本风格差异较大的场景。本文提出一种基于注意力机制的风格表示方法，增加风格特征携带的信息量。文本的文本风格转换模型包括以下步骤：首先对输入句子的词序列与词性序列进行向量化，之后经过两个Bi-LSTM编码器分别计算文本的内容与风格特征序列，将内容序列作用于LSTM解码器生成词汇，而风格序列则经过本文提出的风格调整方法，对输出的词汇概率进行调整，最终输出为指定风格的句子。实验结果表明，对于不同类型的数据，模型的转换准确率与内容保存程度均有更好表现。

关键词

长短期记忆循环神经网络，文本风格转换，注意力机制，序列到序列框架，文本生成

Neural Network Based Text Style Transfer

Zhifeng Hao[1,2], Yusheng Chen[1*], Ruichu Cai[1], Wen Wen[1], Lijuan Wang[1]

[1]School of Computers, Guangdong University of Technology, Guangzhou Guangdong
[2]School of Mathematics and Big Data, Foshan University, Foshan Guangdong
Email: *zfhao@gdut.edu.com

Received: Oct. 8th, 2020; accepted: Oct. 23rd, 2020; published: Oct. 30th, 2020

Abstract

The existing method has simple style representation, and cannot be adapted to datasets with

*通讯作者。

文章引用: 郝志峰, 陈渝升, 蔡瑞初, 温雯, 王丽娟. 基于神经网络的文本风格转换[J]. 计算机科学与应用, 2020, 10(10): 1888-1899. DOI: 10.12677/csa.2020.1010199

large differences in text style. The article proposes a style representation method based on attention mechanism to increase the amount of information carried by style features. The text style conversion model of text includes the following steps: firstly vectorize the word sequence and part-of-speech sequence of the input sentence, and then calculate the content and style feature sequences of the text through two Bi-LSTM encoders respectively, and apply the content sequence to LSTM decoding. The generator generates vocabulary, and the style sequence is adjusted by the style adjustment method proposed in this paper to adjust the output vocabulary probability, and finally output a sentence with a specified style. The experimental results show that the model performs well for different types of data, indicating that the proposed model has good adaptability.

Keywords

LSTM, Text Style-Transfer, Attention, Seq2seq, Text Generation

1. 引言

同样的内容可以用多种不同的方式表达，文本的风格转换即是将句子从当前的表达方式更换到另一种方式，并在这过程中保证描述的内容保持不变。文本风格转换系统能够应用于许多场景，例如 1) 对语言格式进行修饰，协助文员撰写正式文书；2) 将复杂文学名著变得简单易懂，吸引儿童接触阅读；3) 将商品特性转为广告标语，方便商家推广销售。

近些年来，随着 LSTM (Long Short-Term Memory)、GAN (Generative Adversarial Networks)等许多深度学习方法的日渐成熟，如何按照预设期望生成句子也逐渐成为许多人的研究对象[1]。对于如何实现文本风格的转换，目前存在许多不同的研究思路。Xu 等人[2]的方式最为直接，使用一种基于短语的机器翻译系统针对特定的写作风格进行转换，将问题当作翻译问题来处理，并在原始与现代的莎士比亚戏剧文本上进行测试。在此之后，Jang [3]以及 Jhamtani [4]等人分别通过改用基于 Bi-LSTM 的 seq2seq 翻译模型以及加入 coverage 机制来改进转换效果。这类方法将问题视为翻译处理，需要依赖大量平行语料，在推广应用中容易受到数据分布的限制。后续的工作则结合对抗网络将模型应用于非平行语料上，但实验大多是基于“情感”转换。Singh [5]与 Zhang [6]等人就直接使用了无监督机器翻译的方法直接修改文本风格；而文献[7] [8] [9]对风格的表示进行建模，其中 Shen 等人[7]将编码器输出的隐状态序列视为风格特征，而[8] [9]选择在解码时对隐变量加入 one-hot 类别标签，将文本所属类别作为风格特征。

现有的分离“内容”与“风格”的工作，往往使用简单的方式(如固定的向量、标签等)表示风格特征，风格信息存储量少，因而无法表示复杂的风格类型；此外，这类模型采用同一个解码器生成不同风格的文本，要求解码器得同时兼顾两类不同分布的文本数据，因而当不同类别的句子差异较大时，这类模型在转换能力上表现不佳。

针对以上问题，本文提出一种基于神经网络的风格转换模型，分离提取句子的内容与风格信息，借此扩大模型在不同类型风格下的适应程度。模型首先从编码后的句子提取带有上下文信息的风格与内容

特征，借助注意力机制引导调整风格特征，通过风格信息调整输出不同的概率偏好，最终生成出期望风格的句子。之后，参照前人工作，从文本内容保留以及转换能力两方面对模型进行评估，分别在有监督与无监督数据集上进行对比测试。

2. 相关工作

近些年来，国内外有许多关于文本风格转换的研究。从转换方式的角度，这些工作可分为直接端到端以及隐空间编辑两大类：前者构建出从风格 A 向风格 B 单向转换的 seq2seq 模型，而后者通过修改原句的隐状态来实现风格转换。

2.1. 基于翻译框架的文本风格转换

部分研究者从翻译的思路入手，而不讨论风格的表示，如文献[2] [3] [4] [5] [6]就提出使用机器翻译的框架对特定的风格进行转换。这类文章直接将转换问题定义为机器翻译问题，其中源“语言”是待转换的原始文本，目标“语言”即为转换后的文本。其中，Zhang 等人[6]着眼于情感倾向转换，由于数据缺乏平行语料，在训练之前用预训练的语言模型构造伪造的平行语料，借助伪平行数据预训练翻译模型，再通过对抗训练的方式不断优化模型表现；Harsh 等人[4]用 Seq2seq 进行莎士比亚戏剧古今版本的转换，使用双向 LSTM 编码器来充分获取原句子的上下文信息，并加入注意力机制以及 Pointer Network 提高对原句中生僻词汇的关注程度；而 Singh 等人[5]则在编码器与解码器均加入双向 LSTM，将解码器 LSTM 的输出序列直接输入给判别器进行判断；Wang 等人[12]使用 BERT 学习将未处理的原始句子转换为规范化后的句子(如统一大小写，词语纠正等)。

2.2. 修改隐空间的文本编辑

一部分研究者从风格修改的思路着手，将文本对应的标签信息引入模型来协助训练，从而实现对风格变量的建模。这类方法一般对通过修改文本地隐空间表示来实现风格的转换。

Bowman 等人[11]构建了变分自编码 RNN，通过连续地调整文本在隐空间上的表示，实现对文本内容的修改。而 Mueller 等[12]在此基础上，加入分类网络模块的联合近似推断来学习，在该模型下，由于文本在隐空间的表示是连续的，可采用有效的基于梯度的优化来找到附近的局部最优值。通过适当地约束这种优化并使用 VAE 解码器生成调整后的序列，在一定程度上确保修订与原始序列基本相似。Fu 等人[10]提出将编码器输出的隐变量作为内容向量，在训练时加入约束让编码器生成风格无关的隐变量，通过拼接不同的可学习的风格向量实现风格转换。Yi 等人[13]使用两个情感矩阵分别存储不同的情感倾向文本的信息，在编码时借助预先提取的注意力向量让编码器生成情感无关的隐状态，通过不同的情感矩阵控制输出句子的喜恶倾向。

3. 模型框架

3.1. 模型组成

本文提出的模型结构如图 1 所示，该模型包括输入层，编码层，调整层以及解码层。对于输入文本，首先通过嵌入(Embedding)得到包含词意与词性的表示矩阵，然后通过两个双向 LSTM 分别抽取句子的内容与风格信息，之后依据需要，经过调整层抽取修改风格信息，最终由解码层转换为目标风格的文本。现对各个部分进行详细说明。

1) 输入层：消歧的文本编码

输入层将输入的原始文本转为向量表示。相同的词汇在不同的语境下有不同的释义，为了消除歧义，

丰富语句表达，在输入词汇的同时让模型接收词汇对应的词性信息。

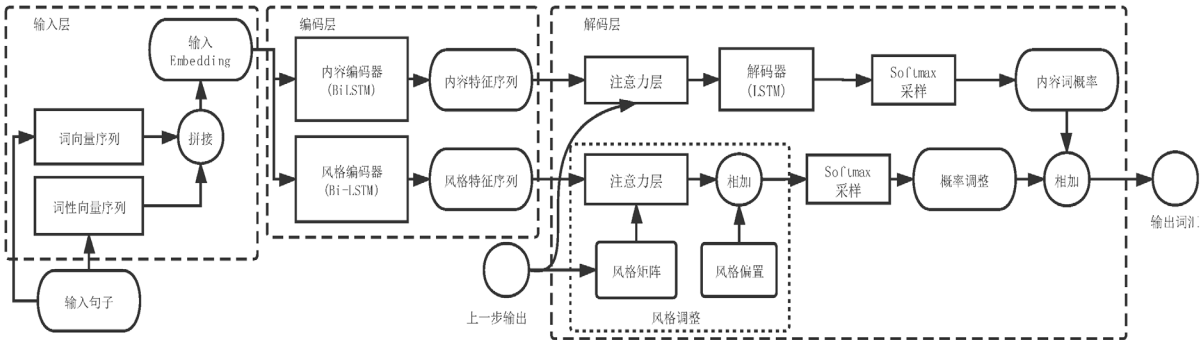

Figure 1. Our text style transfer framework
图 1. 本文的文本风格转换框架

模型的输入包括原始的句子以及对应的词性序列。首先将原始文本的每个单词为 one-hot 编码，通过词嵌入(word embedding)将句子中的每个词将维表示为一个 k 维向量，则每个句子可表示为：$x_w=\left[w_1,w_2,\cdots,w_n\right]^{\mathrm{T}}\in\mathbb{R}^{n\times k}$，其中 w_i 为句中第 i 个词的词向量，n 为句子 x 最大长度。

同样的，句子对应的词性序列经过 one-hot 以及 Embedding 后表示为：

$$x_p=\left[p_1,p_2,\cdots,p_n\right]^{\mathrm{T}}\in\mathbb{R}^{n\times l}$$

其中 p_i 为句中第 i 个词对应词性的高维向量，维度大小为 l。将词序列与词性序列拼接，得到最终消歧后的输入编码。

将输入文字经过编码后输出到下一层，提取风格与内容特征。

$$x=\left[x_w,x_p\right]\in\mathbb{R}^{n\times(k+l)}$$

2) 编码层：依据上下文抽取风格与内容特征

为了修改句子的风格信息，首先经过编码层获取原句子的内容与风格信息。与以往工作不同，本文使用两个编码器分别提取内容与风格特征。

句子的内容与风格的表达需要根据上下文词汇，而为让模型尽可能获取序列上下文信息，采用双向 LSTM 来对提取句子的高维特征。

LSTM 的全称是 Long Short-Term Memory，它是 RNN (Recurrent Neural Network)的一种。相较于传统的 RNN，LSTM 内部引入了记忆单元，因而对序列具有更好的长期记忆功能。如图 2 所示，LSTM 包含输入门、输出门以及遗忘门三个结构，根据上一时刻的隐状态 $h^{(t-1)}$、细胞状态 $C^{(t-1)}$，以及当前时刻的输入 $x^{(t)}$，来计算对历史信息的保留程度。首先计算内部门控，具体如下：

$$i^{(t)}=\sigma\left(W_i\left[h^{(t-1)},x^{(t)}\right]+b_i\right)$$

$$f^{(t)}=\sigma\left(W_f\left[h^{(t-1)},x^{(t)}\right]+b_f\right)$$

$$o^{(t)}=\sigma\left(W_o\left[h^{(t-1)},x^{(t)}\right]+b_o\right)$$

其中 t 时刻的输入门 $i^{(t)}$，遗忘门 $f^{(t)}$，输出门 $o^{(t)}$的计算均是一个单层前馈网络，W，b 分别表示对应的权重以及偏置。之后的序列的状态信息更新方式如下：

$$\hat{C}^{(t)} = \tanh\left(W_c\left[h^{(t-1)}, x^{(t)}\right] + b_c\right)$$

$$C^{(t)} = C^{(t-1)} \odot f^{(t)} + i^{(t)} \odot \hat{C}^{(t)}$$

$$h^{(t)} = o^{(t-1)} \odot \tanh\left(C^{(t)}\right)$$

式子中，$\odot$ 表示向量中对应元素直接相乘；$\hat{C}^{(t)}$ 为新产生的待定细胞状态，由输入门 $i^{(t)}$控制参与更新的维度；遗忘门 $f^{(t)}$对历史信息 $C^{(t-1)}$ 进行取舍，再加上 $\hat{C}^{(t)}$ 更新细胞状态；输出的隐状态 $h^{(t)}$由激活的当前细胞状态经过输出门 $o^{(t)}$筛选后得到。

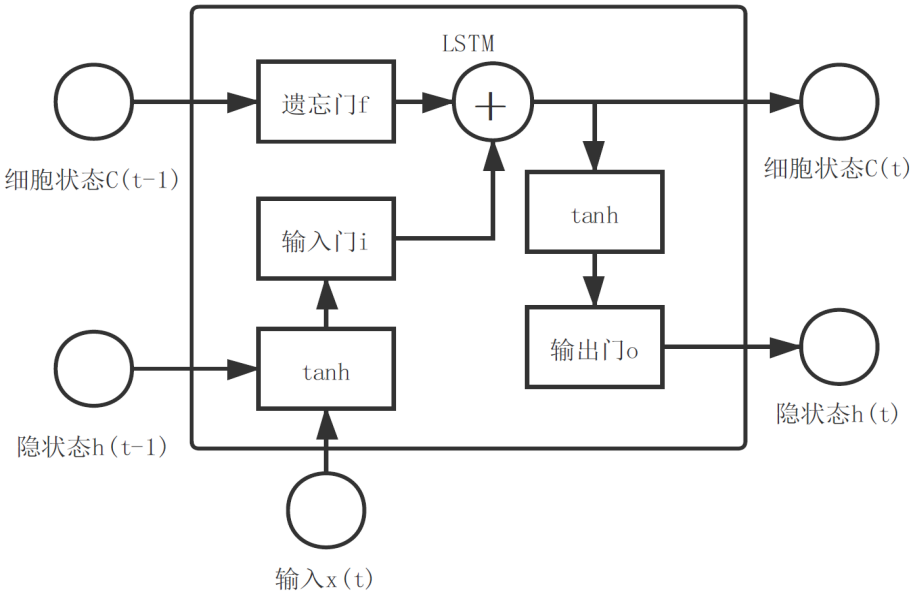

Figure 2. Structure of LSTM
图 2. LSTM 内部结构

LSTM 根据先前状态来输出下一个状态，仅能捕捉先序列的上文信息。双向 LSTM 则能够同时考虑上下文信息，其每一层都中包含了一个前向 LSTM 与一个后向 LSTM，每个时刻的状态由为两个方向的状态信息组合得到，输出的编码能够结合上下文特征。在本文的模型中，t 时刻状态 $h^{(t)}$的计算为前向 $h_L^{(t)}$ 与后向状态 $h_R^{(t)}$ 的相加，即 $h^{(t)} = h_L^{(t)} + h_R^{(t)}$ 。

本文使用两个双向 LSTM 作为编码器，从不同侧面对句子进行编码，以分别提取句子的内容以及风格信息。用 ENC_s 以及 ENC_c 分别表示风格以及内容编码器，$h_s = \left\{h_s^{(1)}, \cdots, h_s^{(n)}\right\}$ 与 $h_c = \left\{h_c^{(1)}, \cdots, h_c^{(n)}\right\}$ 为其提取风格与内容的特征序列。在后续结构中，内容特征 h_c 将被输入到解码器中生成词汇，而风格特征 h_s 则用于调整输出词汇概率分布。

$$h_s = \mathrm{ENC}_s(x); h_c = \mathrm{ENC}_c(x)$$

3) 注意力机制

引入注意力机制的目的在于，利用编码层所有隐状态来构建解码器输出所需的上下文向量。对于内容特征序列，在解码器输出下一个词前，用当前输出的词汇编码 $\hat{x}^{(t-1)}$ 的查询(query)Q 分别与编码器的所有隐状态 $\left\{h_c^{(1)}, h_c^{(2)}, \cdots, h_c^{(n)}\right\}$ 对应的键(key)K 计算相关性，得到各个状态值(value)V 的权重系数。t 时刻关注的文本向量 $c^{(t)}$ 计算如下：

$$a^{(jt)}=\frac{e^{f\left(Q^{(j)},K^{(t)}\right)}}{\sum_j e^{f\left(Q^{(j)},K^{(t)}\right)}}$$

$$c^{(t)}=\sum_j a^{(jt)}V^{(j)}$$

其中，$a^{(jt)}$ 表示与 t 时刻下对第编码序列第 j 个状态的权重系数。K，Q，V 的值均为对应的状态向量 h 经过前馈网络计算得到，$f(*)$ 计算向量之间的相似度。文本向量 $c^{(t)}$ 是隐状态的值的加权平均。

4) 调整层：引导调整风格特征

调整层的目的在于根据预设对原句的风格特征调整。为了让模型能够兼容多种类型风格的文本，风格向量对输出的影响应当具有足够的灵活性，即在每一个时间步，风格特征对生成的词汇的贡献度不同。本文借助注意力机制从 ENC_s 输出的序列抽取风格特征，并在其中根据需求对抽取过程加以不同的引导。首先是查询所用的 key 向量。在注意力机制中，query 与 key 相似，则受关注度越高。在生成不同风格的句子时，模型对风格特征序列 h_s 关注的侧重点应当有所差异，因而采用如下方式得到查询风格所用的键向量 K_s：

$$K_s^{(t)}=\sigma\left(W\hat{x}^{(t-1)}+b\right)^{\mathrm{T}}M_\beta$$

其中 β 为目标风格的编号，每一类风格有其对应的矩阵 $M\in\mathbb{R}^{\alpha\times e}$；每个风格矩阵由 α 个 e 维的属性向量拼接而成，α 为预设种类数量，e 为隐状态的尺寸；设定 K_s 来自矩阵中属性向量的组合，组合方式则是依靠当前输出的词汇编码 $\hat{x}^{(t-1)}$ 经过映射得到，它是一个长度为 α 的向量，每个维度的值均处于[0,1]，表示对 M 中每种属性的权重。

考虑到若目标句子与原句词汇差异较大，仅仅用 attention 的方式难以获得目标风格的特征。因此在经过 attention 抽取风格特征后，依据生成句子的目标风格 β，对其结果添加对应的偏置向量 B_β。当前时刻，调整后的序列特征 $s^{(t)}$计算如下：

$$s^{(t)}=\mathrm{attention}\left(\{h_s\},K_s^{(t)}\right)+B_\beta$$

5) 解码层

解码层的目的是将编码后的特征向量恢复为文本。对隐状态的解码分为内容特征以及风格特征两个部分，内容特征用于句子主干生成，而风格特征负责为对结果进行调整。首先通过不同的方式对二者进行解码，再将结果汇总输出最终的词概率。

文本特征的解码通过一个 LSTM 完成，时刻 t 输出词汇 $\hat{w}^{(t)}$ 的计算过程如下所示：

$$d^{(t)}=\mathrm{LSTM}\left(h^{(t)},\left[c^{(t)},\hat{x}^{(t-1)}\right]\right)$$

$$\hat{w}_c^{(t)}=\mathrm{softmax}\left(\sigma\left(W_{oc}d^{(t)}+b_{oc}\right)/\gamma\right)$$

其中 $\hat{x}^{(t-1)}$ 为上个时刻输出词汇的词向量，$d^{(t)}$为解码器输出向量，$\hat{w}_c^{(t)}$ 是仅基于内容向量输出的词概率；γ 为温度系数，用于在训练时调整概率的平滑程度。

由于风格特征作用在于对输出内容进行微调，为避免出现输出文本过分依赖的局面，仅采用简单的手段对该部分解码：经过一层前馈网络将风格特征映射到词表的长度，后通过 softmax 转为概率。

$$\hat{w}_s^{(t)}=\mathrm{softmax}\left(\sigma\left(W_{os}s^{(t)}+b_{os}\right)/\gamma\right)$$

最后输出的词概率为二者的加权平均，限定内容向量输出的权重 α 高于风格特征的输出，具体大小根据训练数据学习得到。

$$\hat{w}^{(t)} = \alpha \hat{w}_c^{(t)} + (1-\alpha)\hat{w}_s^{(t)}, \quad \alpha \in (0.5,1]$$

3.2. 损失函数

为了让模型能够达到期望的能力，在训练过程中根据应用设定了如下的损失函数。

1) 端到端误差

模型通过不同的风格矩阵引导生成句子的风格类型。当采用与输入文本相同的风格参数时，模型应当类似于一个自编码器，重新构建回原先的文本。对于输入的句子 $x = \left[w^{(1)}, \cdots, w^{(t)}\right]$，采用交叉熵度量此刻的损失，有

$$l_{rec} = -\lambda_{rec} \sum_t \log p\left(P^{(t)}\left(w^{(t)}\right)\right)$$

$$P^{(t)} = \mathrm{DEC}\left(\hat{x}^{(t-1)}, \mathrm{ENC}(x), \theta_a^{(t)}\right)$$

其中 $P^{(t)}$是 t 时刻模型输出的概率向量，当时对应风格特征为 $\theta_a^{(t)}$。

倘若风格 β_1 与 β_2 之间存在平行语料 $(x,y) \in X$，通过优化二者转换之间的交叉熵损失对模型进行训练：

$$l_{ts} = -\lambda_{ts}\left(\sum \log p(y \mid x, \beta_2) + \sum \log p(x \mid y, \beta_1)\right)$$

2) 隐空间对齐

理论上，相同内容的句子，内容编码器的输出内容应当相近，反之则相离。

模型在转换前后，首先应当保证描述内容不失真，因而对于原句 x 以及转换结果 $\hat{x}$，在隐空间对二者内容状态序列的计算差异：

$$l_{c-sim} = f_{\mathrm{dist}}\left(\mathrm{ENC}_c(x), \mathrm{ENC}_c(\hat{x})\right)$$

而对于不同内容的句子对(x_1,x_2)，要使误差，首先经过负梯度层转换后再计算二者距离，

$$l_{c-dis} = f_{\mathrm{dist}}\left(\mathrm{Rev}\left(\mathrm{ENC}_c(x_1)\right), \mathrm{Rev}\left(\mathrm{ENC}_c(x_2)\right)\right)$$

$$l_c = \lambda_c\left(l_{c-sim} + l_{c-dist}\right)$$

而对于风格特征，鼓励增强其灵活性，从而增大对不同类型数据的实用性。因此，不对其计算某一类别的相似度。但对于模型的输入与输出，若是进行了风格的调整，则二者的风格状态序列应当有较低的相似度，反之则相似度高：

$$l_{s-dist} = \lambda_s f_{\mathrm{sim}}\left(\mathrm{ENC}_s(x), \mathrm{ENC}_s(\hat{x})\right)$$

$$l_{s-sim} = \lambda_s f_{\mathrm{dist}}\left(\mathrm{ENC}_s(x), \mathrm{ENC}_s(\hat{x})\right)$$

$$l_s = \lambda_s\left(l_{s-sim} + l_{s-dist}\right)$$

式子中，$f_{\mathrm{dist}}(a,b) = \|a-b\|^2$ 计算矩阵的差异程度，而 $f_{\mathrm{sim}}(a,b) = \cos(a,b)$ 计算向量相似度。

3) 策略梯度

倘若训练数据平行语料较少或不完全可靠(语料由机器生成)，需要引入无监督的训练方式，由风格判别器引导对应风格的文本生成。此时利用策略梯度来计算模型的损失。将新生成的句子 $\hat{x} = \left[\hat{w}^{(1)}, \cdots, \hat{w}^{(n)}\right]$ 视为一个策略序列，需要获得每一个策略 $\hat{w}^{(t)}$ 最终对全局产生的增益。首先通过目标风格语料的 language Model 对已生成序列补全获得完整的序列，对补全的序列通过风格判别模型得到当前策略的增益

$$L^{(t)} = \mathrm{LSTM}\left(\mathrm{Emb}\left(\mathrm{LM}\left(x \mid \hat{w}^{(1)}, \cdots, \hat{w}^{(t)}\right)\right)\right)$$

最终整句生成文本损失大小：

$$l_{pg} = -\lambda_5 \frac{1}{n} \sum_t p\left(\hat{w}^{(t)}\right) \log L^{(t)}$$

4. 实验与分析

4.1. 数据集

为验证所提出模型在不同类型数据的适应性，本文在以下几组数据集上对模型进行训练与测试：古今版本的莎士比亚戏剧，不同翻译版本的圣经，以及 yelp 中不同情感倾向的评论语句。表 1 展示了所使用数据集的统计信息。

圣经译本：Keith 等人[11]搜集了古今多种不同版本的圣经并用于文本风格转换的任务。链接(https://github.com/keithecarlson/StyleTransferBibleData)中为其中对齐后的 8 个公开版本，当中根据版本间的相似程度设定了三个不同难度的数据集合：KJV 到 ASV，King James 译本与美国标准版，BLEU 为 68.72；BBE 到 ASV，基本英语版与美国标准版，BLEU 为 22.75；YLT 到 BBE，Yong 的翻译与基本英语版，BLEU 为 9.42。这三种平行语料共约 18 万条。

莎士比亚戏剧：该数据集首先来自 Wei Xu [2]等人的工作，从网站(http://nfs.sparknotes.com/, http://www.enotes.com/)搜集了共计 23 部莎士比亚戏剧的文本，包括原始英语与现代译本两种版本，并匹配了当中的平行语料，总计达到 4 万条句子。该数据集也在许多风格转换的工作中被使用。

Yelp 评论：许基于多隐空间编辑方法的工作都选择 yelp 评论数据来对模型进行训练测试。本文依据评论对应的打分等级区分文本的情感倾向：打分 4 至 5 星则归为积极类型，2 星及以下归为消极类型的评论。过滤后的句子总数量约在 63 w 以上。由于缺乏平行语料，通过替换反义词的方式为模型构建临时的转换目标。

Table 1. Statistic information of Datasets
表 1. 数据集信息统计

数据集	类别数	句子数量	平均长度	词汇数
莎士比亚	2	42,150	12	19,387
圣经	5	186,416	27	51,529
Yelp 评论	2	638,943	9	10,000

4.2. 对比方法

选取与本文模型结构相似的工作作为基线进行比较：

CE-S2S (Copy-Enriched Sequence to Sequence Models)：Harsh 等人[4]用之间翻译的方式来进行风格转换任务，采用的模型是机器翻译方法里具有代表性的一种。该工作同样采用双向 LSTM 作为模型编码器，LSTM 作为解码器，并在当中加入 attention 机制以及 pointer-network 来增强模型表现。

Style-Embedding：来自 Fu [10]等人的工作，采用隐空间编辑的方法进行转换任务。模型主体由基于 GRU 的编码器与解码器组成，使用对抗训练让编码器生成风格无关的内容向量，通过对内容向量拼接不同的风格向量来改变生成句子的风格。

Ours-SE (双注意力的 Seq2Seq)：在本文模型的基础上，删去风格编码器，文本注意力与风格调整均作用于同一个隐状态序列。

Ours-DE (双编码器的 Seq2seq)：在本文模型的基础上，将调整层替换为用固定的风格 H_β 向量来对风格特征序列做 attention 操作，因此每个时刻对输出的词概率分布调整将相对固定。

4.3. 评估指标

前人的工作[2]-[13]中，大多从风格转换的准确率以及内容保留程度两方面评估模型好坏。

对具有平行语料的数据集，依照 Keith 等人[7]的方法，用 BLEU 与 PINC 分数分别衡量转换结果的准确程度与保留程度：生成句子与目标句子的 BLEU 越高，表示转换准确性越好；而生成句子与原句子的 PINC 越高，代表生成句子创新程度越高，反之则与原句越相似。BLEU 计算如下：

$$\mathrm{BLEU}=\mathrm{BP}\cdot\exp\left(\sum_{n=1}^{N}w_n\log P_n\right)$$

其中，BP(Brevity Penalty)是过短惩罚因子；$\log(P_n)$计算预测句子与参考句子的 n-gram 分数，w_n 是对应 n-gram 分数的权重。

而 PINC 分数的计算如下所示：

$$\mathrm{PINC}\left(\hat{x},x\right)=1-\frac{1}{N}\sum_{n=1}^{N}\frac{\left|\mathrm{Ngram}\left(x,n\right)\cap\mathrm{Ngram}\left(\hat{x},n\right)\right|}{\left|\mathrm{Ngram}\left(x,n\right)\right|}$$

式子中，Ngram(x, n)代表句子 x 的 n-gram 列表，实验中 n 取值在[1,4]之间；PINC 分数本质在于分数是计算原句 x 与生成句子 $\hat{x}$ 计算各种 N-gram 下共有词组数量的占比平均值，当生成句子与原句用词越相似，则分数越低。由于语料之间存在差异性，PNIC 的高低受到目标与原句之间相似性的影响。本文采用相对 PNIC 来计算模型对原句的保留能力：

$$p=\frac{1}{n}\sum\nolimits_i\mathrm{PINC}\left(\hat{x}_i,x_i\right)-\mathrm{PINC}\left(y_i,x_i\right)$$

若 $p>0$，说明模型能够生成更加多样性的用词、句法，$p<0$ 则模型更倾向于保留原句的用词。

在无监督语料中，为了评估模型是否顺利将文本从一种风格转移到另一种风格，最直观的方法是对生成文本的风格所属进行判断。Zhirui [5]等人采用一个文本分类器为判断文本的转换结果，且在实验中，分类器对模型的优劣判断与人类评分的结果相似。因而在测试 yelp 数据集中，本文采用具有相同结构的预训练分类器，对各模型的风格转换能力进行评估。对于内容的一致程度，则通过计算生成句子与原句子的 BLEU 分数来反映。

4.4. 参数与设置

将数据集按照 8:1:1 划分为训练集、验证集以及测试集。对于 yelp 与 Shakespeare，句子按照 20 的最大长度对尾部进行截断，并筛去长度小于 4 的文本；而圣经文本则截去超过 35 个词的部分。将英文文本统一小写，之后采用 Keras 提供的 tokenizer 库对文本进行分词与语料搭建，将词频低于 5 的词替换为“<unk>”。之后根据词库 id 替换单词文本，将句子转为 one-hot 编码序列。此外，通过 NLTK 库获取原始输入句子的词性序列，构建对应的语料库，转换为词性 id 序列。

关于模型参数，词向量与词性向量大小设为 128，双向 LSTM 以及 LSTM 隐空间大小为 256，均采用单层结构；训练过程中初始的温度系数 γ 为 100，batch size 为 16 对共 32 条句子。模型参数的优化采用 ADAM 算法，使用 Pytorch 默认参数。经过测试，在莎士比亚与圣经数据集中，各个损失函数的权重 $\lambda_{rec}=0.5$，$\lambda_{ts}=1.0$，$\lambda_s=0.2$，$\lambda_c=0.2$，$\lambda_{pg}=0.1$；而在 yelp 中，损失权重设为 $\lambda_{rec}=0.6$，$\lambda_s=0.2$，$\lambda_c=0.2$，$\lambda_{pg}=0.5$。

4.5. 实验结果与分析

1) 内容保留与风格程度

分别在不同数据集上对各模型进行测试。首先在无监督语料下测试各模型的内容保持与转换成功率，结果如表 2 所示。从整体上看，本文转换模型在内容保留与转换程度上均高于基线模型 StyleEmbedding，且完整结构下(双编码器 + 动态调整层)的表现最优，这充分说明模型具有良好的转换能力与保存内容的能力。其次，采用双编码器(Ours, Ours-DE)分开提取内容与风格，相比于单编码器模型(StyleEmbedding, Ours-DA)，能够更好地保存内容详细；另外，从风格转换上，本文提出的风格调整方式相比于固定的风格向量(Ours-DE, Style-Embedding)具有更好的转换效果。

Table 2. Model performance for unsupervised datasets
表 2. 无监督数据下的模型表现

模型	内容保存(BLEU)	转换程度(ACC)
StyleEmbedding	21.10	87.34%
Ours-DA	22.92	92.06%
Ours-DE	23.31	91.69%
Ours	**25.68**	**92.34%**

表 3 是模型再有监督数据下的表现，与无监督数据有着相似的结果：在莎士比亚古今版本以及各类别圣经的转换任务中，模型的 BLEU 值均优于基线 CE-S2S (S2S + attention + pointer)，说明具有更好的转换能力；另外，本文模型的相对 PINC 值普遍低于基线模型，反映其在句子生成上相对保守，即更加倾向于对原句进行少量改动，保持原有的词汇组合。

2) 适应能力讨论

Table 3. Model performance for supervised datasets
表 3. 有监督数据下模型表现

数据	模型	目标转换 BLEU	内容保留 P(PINC)
Shakespeare Original & modern	CE-S2S	30.25	35.73%
	Ours-DA	32.46	28.98%
	Ours-DE	32.51	24.13%
	Ours	**34.14**	20.21%
Bible KJV & ASV	CE-S2S	64.43	15.85%
	Ours-DA	65.23	0.73%
	Ours-DE	67.96	-0.57%
	Ours	**68.41**	-0.11%
Bible BBE & ASV	CE-S2S	30.01	6.53%
	Ours-DA	33.62	5.87%
	Ours-DE	34.41	5.03%
	Ours	36.01	-4.25%
Bible YLT & BBE	CE-S2S	22.18	14.17%
	Ours-DA	23.01	14.18%
	Ours-DE	23.45	12.89%
	Ours	**23.92**	12.64%

借助不同相似性(BLEU)的圣经数据代近似代表不同类型风格下的文本。表 4 显示各类型数据下，模型对风格信息的自适应调整：当两类风格之间的语句相似性高时(KJV & ASV)，风格信息需求较少，内容特征提供了 0.82%增益；而随着句子差异增大，模型需要更多的风格信息来协助。为直观展示不同模型对生成的影响，在表 5 中对各模型部分转换结果进行展示。

Table 4. Weights learned from data with different similarities
表 4. 不同类型数据下的内容概率权重

	KJV & ASV	BBE & ASV	YLT & BBE
α	0.822	0.719	0.510

Table 5. Transferred sentences by different model
表 5. 各数据集转换结果示例

数据来源	模型	实例句子	BLEU	PINC
Yelp		I will never go here again.		
	styleEmbedding	i will continue go here again .	-	-
	Ours	i will definitely go here again .	-	-
Shakespeare	Source	Thou chid'st me oft for loving Rosaline.		
	CE-S2S	you chid'st me often for loving rosaline .	36.56	5.00
	Ours	you scolded me often for loving rosaline .	68.04	12.74
	Target	You scolded me often for loving Rosaline.		
Bible KJV & ASV	Source	I thank my God upon every remembrance of you,		
	CE-S2S	i thank my god upon all my remembrance of you ,	100	0.00
	Ours	i thank my god upon all my remembrance of you ,	100	0.00
	Target	I thank my God upon all my remembrance of you,		
Bible BBE & ASV	Source	Be on the watch against dogs, against the workers of evil, against those of the circumcision:	-	-
	CE-S2S	be on the watch of the dogs, beware the evil workers, beware those of the circumcision :	32.23	−33.48
	Ours	beware of the watch of the dogs, beware of the evil workers, beware of the circumcision :	66.90	−13.97
	Target	Beware of the dogs, beware of the evil workers, beware of the concision:	-	-
Bible YLT & BBE	Source	and Saul saith, `Hear, I pray thee, son of Ahitub;' and he saith, `Here I, my lord.'		
	CE-S2S	and saul said , hear , I pray you , son of ahitub . and he said that , here I am , my lord	35.30	−14.36
	Ours	and saul said , hear , I pray you , son of ahitub . and answering he said , here I am , my lord	48.39	−13.11
	Target	and Saul said, Give ear now, O son of Ahitub. And answering he said, Here I am, my lord.		

5. 结束语

本文提出了一种基于循环神经网络的风格转换方法。在传统 seq2seq 框架上，通过两个双向 LSTM 从不同角度提取序列信息，以此获取句子的内容与风格特征；并引入风格矩阵与风格偏置，对输出概率实时调整，从而提高风格向量的表达能力，模型对不同风格的适应性。在不同类型数据集的实验表明，对比机器翻译与隐向量编辑等方法，本文的方法在转换能力与内容保留程度上均具有明显的优势。下一

步考虑引入聚类方法表示风格，增强风格变量的可解释性。

基金项目

国家自然科学基金(61472089)；NSFC-广东联合基金(U1501254)；广东省自然科学基金资助项目(2014A030308008)；广东省科技计划项目(2015B010108006)。

参考文献

[1] Kabbara, J. and Cheung, J.C.K. (2016) Stylistic Transfer in Natural Language Generation Systems Using Recurrent Neural Networks. *Proceedings of the Workshop on Uphill Battles in Language Processing*: *Scaling Early Achievements to Robust Methods*, Austin, TX, November 2016, 43-47. https://doi.org/10.18653/v1/W16-6010

[2] Xu, W., Ritter, A., Dolan, B., *et al*. (2012) Paraphrasing for Style. *International Conference on Computational Linguistics*, 2899-2914.

[3] Jang, S.W., Min, J. and Kwon, M. (2017) Writing Style Conversion using Neural Machine Translation.

[4] Jhamtani, H., Gangal, V., Hovy, E., *et al*. (2017) Shakespearizing Modern Language Using Copy-Enriched Sequence-to-Sequence Models. *Entropy* (*Type. Dist*), **6**, 6.06.

[5] Singh, A. and Palod, R. (2018) Sentiment Transfer Using Seq2Seq Adversarial Autoencoders. arXiv: Computation and Language.

[6] Zhang, Z., Ren, S., Liu, S., *et al.* (2018) Style Transfer as Unsupervised Machine Translation. arXiv: Computation and Language.

[7] Shen, T., Lei, T., Barzilay, R., *et al*. (2017) Style Transfer from Non-Parallel Text by Cross-Alignment. *Advances in Neural Information Processing Systems*, 6830-6841.

[8] Fu, Z., Tan, X., Peng, N., *et al.* (2018) Style Transfer in Text: Exploration and Evaluation. *National Conference on Artificial Intelligence*, 663-670.

[9] Yamshchikov, I.P., Shibaev, V., Nagaev, A., *et al.* (2019) Decomposing Textual Information for Style Transfer. arXiv: Computation and Language. https://doi.org/10.18653/v1/D19-5613

[10] Wang, Y., Wu, Y., Mou, L., *et al*. (2019) Harnessing Pre-Trained Neural Networks with Rules for Formality Style Transfer. *Empirical Methods in Natural Language Processing*, 3571-3576.

[11] Bowman, S.R., Vilnis, L., Vinyals, O., *et al.* (2015) Generating Sentences from a Continuous Space. arXiv preprint arXiv:1511.06349. https://doi.org/10.18653/v1/K16-1002

[12] Mueller, J., Gifford, D. and Jaakkola, T. (2017) Sequence to Better Sequence: Continuous Revision of Combinatorial Structures. *International Conference on Machine Learning*, 2536-2544.

[13] Zhang, Y., Sun, X., Xu, J., *et al.* (2018) Learning Sentiment Memories for Sentiment Modification without Parallel Data. *Proceedings of the* 2018 *Conference on Empirical Methods in Natural Language Processing*, Brussels, 1103-1108. https://doi.org/10.18653/v1/D18-1138