计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2020,29(9):136−141 [doi: 10.15888/j.cnki.csa.007433]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于矩阵变换的文本风格迁移方法①

黄若孜[1,2], 张　谧[1,2]

[1](复旦大学 软件学院, 上海 201203)
[2](复旦大学 上海市智能信息处理重点实验室, 上海 201203)

摘　要: 文本风格迁移一直是自然语言处理 (NLP) 中的一个研究热点, 近年来, 随着文本生成方法的发展, 越来越多的工作着眼于不成对 (non-parallel) 文本风格迁移这一任务. 这一任务的目标是, 利用不包含一一对应句子的两个或多个不同风格的文本集, 学习一个迁移模型, 实现改变句子的风格的同时保留句子其他的内容. 目前针对该任务, 已有一些基于生成对抗网络的迁移算法被提出, 但是受限于对抗学习本身的训练不稳定, 以及对句子的风格和语义的独立性假设本身不合理, 这些方法无法高效的学到迁移效果好的模型. 在这篇文章中, 我们首次从统计学习的角度给出了文本风格的定义—文本集中语义向量的协方差矩阵, 在这种新的观点下, 文本的风格依赖于所有句子的语义向量. 我们随后提出了一种无学习 (learning free) 迁移方法, 我们只需要预训练一个自编码器来得到句子的语义向量, 然后对这些向量进行白化和风格化变换, 来实现风格迁移.

关键词: 自然语言处理; 表示学习; 文本风格迁移

引用格式: 黄若孜,张谧.基于矩阵变换的文本风格迁移方法.计算机系统应用,2020,29(9):136–141. http://www.c-s-a.org.cn/1003-3254/7433.html

Text Style Transfer Based on Matrix Transformation

HUANG Ruo-Zi[1,2], ZHANG Mi[1,2]

[1](Software School, Fudan University, Shanghai 201203, China)
[2](Shanghai Key Laboratory of Intelligent Information Processing, Fudan University, Shanghai 201203, China)

Abstract: Text style transfer is always a hot spot in Natural Language Processing (NLP). In recent years, as the development of sequence generation methods, many researchers focus on style transfer on non-parallel corpora. Specifically, this task wants to change the style of the sentence while keeping the original content. To achieve this target, many works have been proposed which based on the generative adversarial network. But due to the instability of adversarial training and the limitation of the independence assumption between the style and semantic information, these methods are hard to learn an effective and efficient transfer model. In this study, motivated by statistic learning methods, a definition of the text style is given. The style of the corpus can be captured by the covariance matrix of its sentences' semantic vectors. From this perspective, the text style is dependent on all the semantic information. We then propose a learning free transfer method where the only thing we need is a pre-trained auto-encoder to produce the semantic vectors. With a pair of matrix transformations, including whitening transformation and stylizing transformation, performing on these vectors, we achieve text style transfer.

Key words: Natural Language Processing (NLP); representation learning; text style transfer

近年来, 文本风格迁移是自然语言处理中的一个热点, 该领域深刻影响着很多 NLP 应用的发展, 比如在生成诗歌的任务中, 利用风格迁移的方法来生成不同风格的诗歌[1]. 文本风格迁移的目标是将原文本重写

① 收稿时间: 2019-10-27; 修改时间: 2019-11-20; 采用时间: 2019-12-05; csa 在线出版时间: 2020-09-04

成其他的风格的新文本, 新文本应该流畅逼真, 同时保留原文本中与风格无关的其他的信息. 举例来说, 从 Yelp 数据集中可以拿到用户对餐厅的评论, 我们希望将这些评论从正面改为负面, 此时文本的风格即为评论中包含的态度. 对于这个任务, 如果有评论的内容一一对应而态度相反的两组文本集, 我们可以很容易的设计一个 Seq2Seq 的模型、有监督的进行训练. 然而, 在大部分风格迁移的场景中, 这样的数据集是缺失的, 于是很多研究者选择利用没有成对句子的数据集学习文本风格迁移的模型.

在已有的文本风格迁移的研究中, 绝大部分工作都认为一个文本集的风格是一个认为给定的标签, 而不是从文本集中自动提取的表示. 比如对于 Yelp 数据集, 评分大于 (小于)3 的评论被认定为正面 (负面) 态度. 从这个观点出发, 一种常见的思路是学习句子与风格无关的语义表示, 然后利用这个表示和另一种风格的标签恢复出句子[2–4], 这实际上是假设句子包含的语义信息和风格信息是相互独立的. 具体来说, 这些工作利用自编码器将不同的风格的句子压缩到一个共享的语义空间, 并将来自不同风格文本集的语义表示的分布进行对齐, 这可以通过附加一个分类器实现, 分类器试图区分出句子的域, 而自编码器试图骗过这个分类器, 经过对抗的训练, 最终使学到的语义表示不包含风格标签的信息.

虽然上述基于对抗训练的模型可以取得一定的效果, 但是正如文献 [5,6] 所指出, 这些工作难以同时改变风格并且保留其他的语义信息. 这些观察表明, 由于句子中风格和语义信息是以复杂的方式混杂在一起的, 独立性假设可能是不合理的. 此外, 基于对抗训练的方法往往收敛缓慢, 非常耗时.

在本文中, 对于文本的风格, 我们提出了新的观点: 如果可以将句子的全部语义信息压缩到一个连续空间中, 则得到的向量包含了该句子全部的语义信息. 那么一组句子的风格, 可以被其对应语义向量的高阶统计量所捕捉. 在具体的实验设置中, 我们选择了协方差矩阵来捕捉文本的风格. 图 1 中, 我们将 Yelp 中的评论按照对应的评分划分成小的文本集, 并且分别计算其对应的协方差矩阵, 然后我们对得到的四个矩阵进行特征值分解, 保留了各自前 50 维特征向量绘制成热力图. 可以看到, 随着评论态度的变化, 得到的特征向量的颜色呈规律性渐变. 这说明了协方差矩阵确实能够捕捉文本风格, 甚至可以区分出风格的强度.

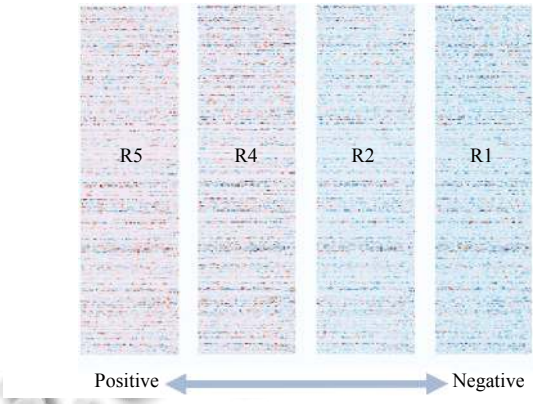

图 1 Yelp 对应不同评分的文本子集的协方差矩阵

基于这一观点, 我们提出了一种半监督学习的方法将句子映射到一个连续空间, 使得来自不同文本集句子的连续表示是可分的. 之后我们提出了一对矩阵变换的算子, 将语义向量先后经过白化和风格化变换以实现风格迁移.

后文组织如下: 第 1 节介绍如何获得文本集中风格的表示; 第 2 节介绍白化-风格化迁移算法; 第 3 节介绍实验设置以及实验结果; 第 4 节进行总结各前 50 维特征向量的热力图. 其中 R5 对应最正面的评价的文本集, R1 对应最负面的评价的文本集.

1 获取文本中的风格信息

1.1 句子的语义向量

对于句子 $x=\{x_1,x_2,\cdots,x_n\}$, 其中 x_i 是词表中的单词, 我们首先需要将句子嵌入到连续空间. 虽然目前已有很多成熟的句嵌入算法[7,8], 但是为了便于进行后续的风格迁移工作, 我们需要该空间满足以下两个条件:

(1) 句嵌入应该是无损的、可以被重建的.

(2) 不同文本集得到向量应该是可分的.

其中第一点意味着该向量需要包含句子的全部信息, 以便于我们从向量中恢复出原始的句子; 第二点意味着, 来自不同文本集的句子在该空间构成的分布应该尽可能不重叠.

我们首先利用映射函数 $E: x \to z \in R^d$, 将句子从原始离散空间映射 d 维连续空间. 为了满足条件 (1), 我们需要一个逆向的映射函数 $D: z \to y$, 重建文本 $y=\{y_1,y_2,\cdots,y_n\}$ 满足 $y=x$, 这两组映射构成一个自编码器. 由于原始文本

和重建文本都是离散的序列, 我们选择用常见的 Seq2Seq 模型[9] 来实现这个自编码器. Seq2Seq 模型是一种基于循环神经网络 (RNN), 对于成对的序列通过最小化重建误差进行端到端训练的模型, 比如机器翻译任务中, 有英文和法文一一对应的训练集, 就可以使用 Seq2Seq 模型进行训练. 使用 Seq2Seq 模型可以选择不同的循环单元, 在本文的实验部分, 我们验证了使用长短期记忆网络 (LSTM), 门控循环单元网络 (GRU), 以及双向的 GRU 等都可以很好的获取文本的风格. 下面我们以 LSTM 为例, 对自编码器的训练目标进行说明.

用作编码器的循环单元记为$LSTM_E$, 第i次迭代得到的隐变量记做h_i, 经过下式所示编码过程, 原始句子被映射为语义向量 z:

$$\begin{cases} h_i = LSTM_E(x_i, h_{i-1}) \\ z = LSTM_E(x_n, h_{n-1}) \end{cases}$$

用作解码的循环单元记做$LSTM_D$, 得到的隐变量记做s_i:

$$\begin{cases} s_1 = LSTM_D(x_1, z) \\ \quad \vdots \\ s_i = LSTM_D(x_i, s_{i-1}) \end{cases}$$

为了从隐变量预测出具体的词, 可以经过一个全连接层之后再做 Softmax 变换, 这样就得到了词表V中各个词在当前位置i出现的概率, 从这个概率中采样单词来生成句子:

$$p(y_i|s_i) = Softmax(W_1 s_i)$$

其中, $W_1 \in R^{|V|\times d}$.

自编码器的目标是重建出输入的文本, 可以通过最小化下式交叉熵损失函数来实现:

$$l_{\text{res}} = -\sum_{x\in X}\sum_{i}\log(p(x_i|s_i))$$

我们再来考虑第二个句嵌入的条件, 为了使得来自不同文本集的向量的分布尽可能不重叠, 可以加入一个分类器来半监督的训练. 假设有两个文本集X_1和X_1, 我们利用同一个自编码器来将两组文本映射到同一个d维空间, 在这个空间中我们定义了如下分类器:

$$\begin{cases} p(t=1|z) = Sigmoid(W_2 f(zz^{\text{T}})) \\ p(t=0|z) = 1 - p(t=1|z) \end{cases}$$

其中, f代表 flatten 函数, 作用是将矩阵展平成一个向量; $W_2 \in R^{d^2}$是一个线性变换, 可以产生一个标量, 这个标量经过 Sigmoid 函数得到一个概率值, 表示该句子来自第一个文本集的可能性. 我们为两个文本集的文本标定标签 t 为 0 或 1, 然后利用如下目标函数进行训练:

$$l_{\text{cls}} = -\sum_{z_i}(t_i\log(p(t_i=1|z_i)) + (1-t_i)\log(p(t=0|z_i)))$$

最终的训练目标为:

$$L = l_{\text{res}} + \alpha l_{\text{cls}}$$

其中, α是调节两部分相对权重的超参数.

1.2 文本集的风格

在日常的生活中, 人们总是可以直观感受到不同文本的风格差异. 在很多语言学的文献中[10], 也已经有了一些成熟的理论来描述生活中的风格现象. 为了得到机器可以理解的文本风格的表示, 我们先给出该表示需要满足的性质, 这些性质与我们的经验是一致的.

首先, 文本的风格是一种统计现象. 单一句子无法形成一种“风格”, 而包含多个句子的文本集中蕴含着人类可以辨别的风格. 因此我们要学到的表示是对文本集整体而言的. 第二, 文本的风格蕴含在文本的语义中. 举例来说, 对于评论“The food is awful and I will not come again.”来说, 如果认为情绪是一种风格, 我们很难区分“not come”是句子的语义部分还是风格部分, 这两部分信息往往不是相互独立的. 最后, 文本的风格是有不同强度的, 如果将上面的句子改为“The food is so awful and I will never come again.”, 尽管都是负面的评价, 蕴含的情绪会比原句更强烈一点.

为了满足这些性质, 我们提出了用文本集语义向量的协方差矩阵来捕捉文本的风格. 对于有 N 个句子的文本集, 假设所有语义向量构成的集合为$Z = [z_1, z_2, \cdots, z_N] \in R^{d\times N}$, 则协方差矩阵为:

$$S = ZZ^{\text{T}}/(N-1)$$

为了验证该矩阵是否能够捕捉文本的风格, 我们用 Yelp 数据集中评分为 1、2、4、5 的评论各自构成文本集, 取这 4 个文本集的协方差矩阵的前 50 维特征向量, 进行了可视化. 图 1 表明, 我们提出的表示确实可以捕捉文本集的风格, 甚至可以区分风格的强度.

2 一种无学习的风格迁移方法

为了利用我们提取的风格表示控制文本集的风格, 我们提出了一种基于矩阵变换的文本风格迁移方法: 如果语义向量的协方差矩阵可以代表文本集的风格, 我们可以直接将另一个文本集的协方差调整至和该文

本集相同, 从而实现风格的“对齐”.

由于协方差矩阵是半正定的, 可以进行特征值分解:

$$S = P\Lambda P^{\mathrm{T}}$$

其中, Λ是由S特征值构成的对角矩阵, P是由对应的特征向量构成的正交矩阵

如果有两个文本集X_1和X_2, 根据第 1 节的方法, 我们可以得到各自的风格表示S_1和S_2. 现在想要将第二个文本集的风格迁移成和第一个文本集相同, 为此我们对第二个文本集的语义向量组Z先后进行如下矩阵变换:

ZCA 白化: 白化变换会拆除向量各维度之间的相关性, 经过白化之后的向量协方差矩阵为单位矩阵:

$$Z' = P_2\Lambda_2^{-\frac{1}{2}}P_2^{\mathrm{T}}\left(Z - \hat{z}_2 1_d^{\mathrm{T}}\right)$$

其中, $\hat{z}_2$是向量组Z的均值向量, 白化变换之后得到Z'满足$Z'Z'^{\mathrm{T}} = I$.

风格化: 风格化是白化的逆变换, 可以按照第一个文本集的风格重新建立各维度直接的相关性.

$$Z'' = P_1\Lambda_1^{\frac{1}{2}}P_1^{\mathrm{T}}Z' + \hat{z}_1 1_d^{\mathrm{T}}$$

其中, $\hat{z}_1$是文本集X_1向量组的均值向量, 风格化变换之后得到Z''满足$Z''Z''^{\mathrm{T}} = S_1$.

上式展示了如何将文本集X_2迁移为X_1的风格, 反之亦然. 整个迁移的过程不需要训练一个端到端的网络结构, 只需要进行矩阵变换. 将变换后的向量用 1.1 节定义的解码器进行解码, 即得到了迁移之后的句子.

整个模型结构见图 2, 两侧方框表示文本集, 梯形框表示需要训练的网络结构, 中心方框表示实现风格迁移的白化-风格化变换算子. 整个算法步骤如下:

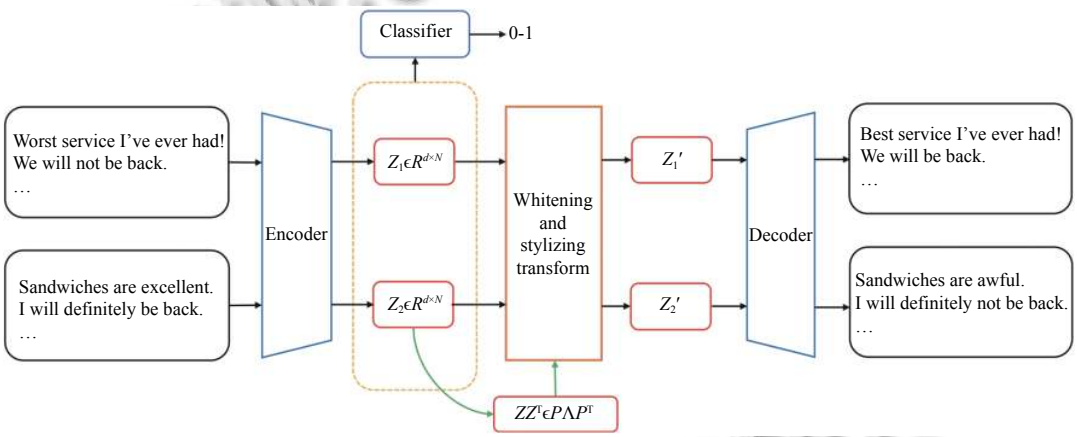

图 2 无学习的风格迁移方法

算法 1. 白化-风格化算法

(1) 预训练: 对于左侧情绪分别为正\负的两个文本集, 利用自编码器进行重建, 同时在隐层语义空间中利用一个分类器半监督的训练, 从而调整该空间的分布.

(2) 获取风格表示: 预训练收敛以后, 用得到的语义向量, 得到两个文本集的语义协方差矩阵.

(3) 风格迁移: 利用白化-风格化变换算子将两个文本集的语义向量进行风格迁移, 迁移后的向量利用已经训练好的解码器进行解码.

3 实验

3.1 实验设置

我们使用了 Yelp 数据集进行了实验, 该数据集收集了 Yelp 网站上用户对餐厅的评价. 其中每一句评价都与一个从 1 到 5 的分值相关联, 分值越高意味着该评价越正面. 我们将该分值超过 3 的作为情绪正面的文本, 低于 3 的作为情绪负面的文本, 从而得到一组风格对立的文本集, 在这一对文本集上, 我们利用提出的白化-风格化算法进行风格迁移. 经过上述处理的 Yelp 文本集的词表大小为 9603, 训练集中包含 173 000 个情绪正面的句子和 26 300 个情绪负面的句子, 验证集中分别包括 37 614 和 24 849 个句子, 测试集分别包括 76 392 和 50 278 个句子.

我们使用了以下两个基于对抗训练的基线模型:

CrossAligned[3]: 该模型假设不同风格的文本集存在一个共享的、与风格无关的语义空间, 该模型通过对抗的训练来对齐不同文本集在这个空间的分布, 以达到去除风格信息的目的.

StyleEmbedding[2]: 该模型显式地学习了不同风格

的嵌入, 将风格嵌入和语义向量一起作为解码器的输入, 从而对于多种风格, 只需要一个自编码器.

本文使用了以下两个指标来评估模型在不成对文本风格迁移上的表现:

Accuracy: 为了评估生成的文本是否符合预期的风格, 我们首先在训练集上预训练了一个文本风格的分类器, 该分类器使用 TextCNN 模型[11], 在测试集上的分类准确率可达到 97.23%. 我们用该分类器对迁移之后的文本的分类准确率作为评估指标, 也就是说, 迁移之后的文本越多可以“骗过”风格分类器, 在这一指标的表现越好.

BLEU: 为了评估生成的文本在改变了风格的同时是否保留了源文本的内容信息, 我们以源文本为参考文本计算了累积 4-gramBLEU 值. BLEU 越高, 意味着和源文本更加相似.

正如文献 [5,6] 提出的, 这两个指标之间往往呈负相关. 直观来看, 成功的改变句子的风格会不可避免的降低 BLEU 值. 所以我们需要一个综合指标来判断模型的效果, 由于两个指标的区间相同, 我们简单的取均值来评估, 其他的综合评估的方法留待之后的工作探索.

我们还考察了不同模型的效率, 用不同模型的训练时间来评判. 所有实验都在同一个 Linux 服务器上运行, 该服务器搭载 Ubuntu 16.04 系统, 使用 Intel theanoXeon E5-2620 v4 的 32 核处理器和两块 NVIDIA GeForce GTX 1080 显卡.

3.2 实验结果

我们先使用 CBOW 算法[12] 将词表嵌入到一个 300 维的连续空间中, 然后固定学到的词嵌入, 利用我们提出的白化-风格化迁移算法在多种循环单元上进行了实验. 用到的网络结构包括 300 维的 LSTM(WS-LSTM), 300 维的 GRU(WS-GRU), 以及两个方向各 150 维的双向 GRU(WS-BiGRU), 我们也实验了引入了注意力机制的效果 (WS-attention). 对所有的结构, 在预训练阶段超参数α都设置为 1, 训练 50 个 epochs, 此外我们还设置了一个没有监督信息的对照组 (WS-unsupervised), 超参α设置为 0; 在之后的阶段, 除了 LSTM 拼接了隐层状态和单元状态得到了 600 维的语义向量, 其他结构都使用 300 维的隐层状态作为句子的语义向量.

将基线模型和基于不同结构的白化-风格化算法的表现展示为图 3. 可以看到本文设置的对照组 WS-unsupervised 在保留语义内容上效果特别好, 但是风格迁移的能力较差, 这是因为在预训练阶段没有引入风格类别的监督信号, 这样学到的语义空间不满足我们提出的第二个条件, 不同文本集得到语义向量不可分. 在引入了监督信号后, Accuracy 得到的巨大的提高, 同时 BLEU 值有所下降, 综合表现优于 WS-unsupervised. 注意到, 此时各种网络结构下我们的模型都是优于两个基线模型的, 除了 WS-attention, 这是因为引入注意力机制后, 得到的语义空间不满足我们提出的第一个条件, 语义信息不是无损的嵌入到这个空间里, 很多信息是由编码器直接提供给了解码器.

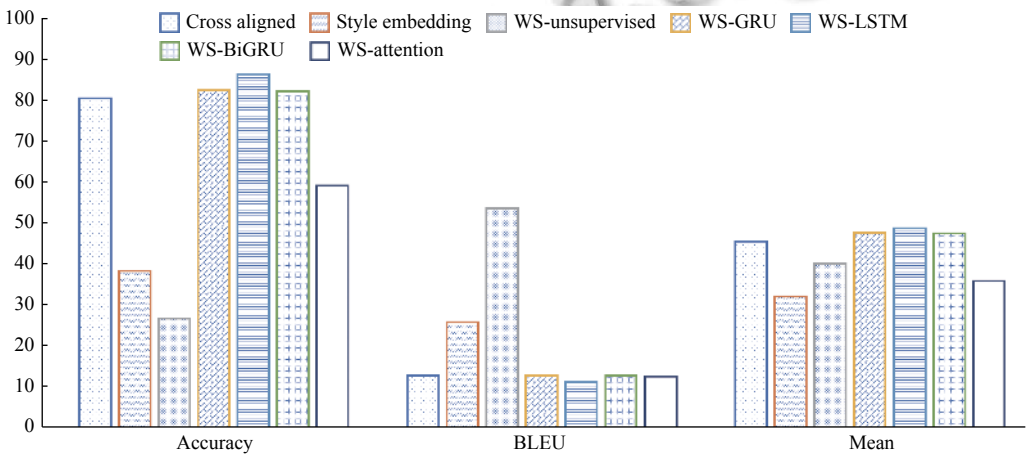

图 3 实验结果

表 1 展示了不同模型训练到收敛的时间, 可以看到, 我们的模型只在预训练阶段需要端到端的学习一个自编码器, 风格迁移的阶段是不需要学习的, 整体效率远远高于基于对抗训练的基线模型.

表 1 不同模型的训练时间 (单位: s)

模型	Cross Aligned	Style Embedding	WS-LSTM
时间	20812	27562	8027

4 结语

本文提出了一种从文本集中提取风格信息的方法, 即将句子嵌入到连续的语义空间, 利用这些语义向量的协方差矩阵来捕捉风格. 利用提取到的风格表示, 我们进一步提出了一种基于矩阵变换的风格迁移方法, 即白化-风格化算法. 实验表明, 该算法的效率远远高于基线模型, 同时迁移的效果也更好.

参考文献

1 Yan R. I, poet: Automatic poetry composition through recurrent neural networks with iterative polishing schema. Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI-16). 2016. 2238–2244.

2 Fu ZX, Tan XY, Peng NY, *et al*. Style transfer in text: Exploration and evaluation. Proceedings of the 32nd AAAI Conference on Artificial Intelligence. New Orleans, LA, USA. 2018.663–670.

3 Shen TX, Lei T, Barzilay R, *et al*. Style transfer from non-parallel text by cross-alignment. Advances in Neural Information Processing Systems 30. Long Beach, CA, USA. 2017. 6830–6841.

4 Hu ZT, Yang ZC, Liang XD, *et al*. Toward controlled generation of text. Proceedings of the 34th International Conference on Machine Learning. Sydney, NSW, Australia. 2017. 1587–1596.

5 Li JC, Jia RB, He H, *et al*. Delete, retrieve, generate: A simple approach to sentiment and style transfer. arXiv: 1804.06437, 2018.

6 Xu JJ, Sun X, Zeng Q, *et al*. Unpaired sentiment-to-sentiment translation: A cycled reinforcement learning approach. arXiv: 1805.05181, 2018.

7 Arora S, Liang YY, Ma TY. A simple but tough-to-beat baseline for sentence embeddings. Proceedings of ICLR 2017. Toulon, France. 2017.

8 Kiros R, Zhu YK, Salakhutdinov R R, *et al*. Skip-thought vectors. Proceedings of the 28th International Conference on Neural Information Processing Systems. Montreal, QC, Canada. 2015. 3294–3302.

9 Sutskever I, Vinyals O, Le Q V. Sequence to sequence learning with neural networks. Proceedings of the 27th International Conference on Neural Information Processing Systems. Montreal, QC, Canada. 2014. 3104–3112.

10 Bell A. Language style as audience design. Language in Society, 1984, 13(2): 145–204. [doi: 10.1017/S0047404500001037X]

11 Kim Y. Convolutional neural networks for sentence classification. arXiv: 1408.5882, 2014.

12 Mikolov T, Sutskever I, Chen K, *et al*. Distributed representations of words and phrases and their compositionality. Proceedings of the 26th International Conference on Neural Information Processing Systems. Lake Tahoe, NV, USA. 2013. 3111–3119.