



基于Point-Then-Operate方法的 迭代写作风格迁移研究

2021春季学期模式识别大作业



清华大学
Tsinghua University

CONTENTS

目录

1/ 文献调研

2/ 文献复现

3/ 短篇小说数据集

4/ 程序项目结构

5/ 模型训练

6/ 分析与总结



清华大学
Tsinghua University

CONTENTS

目录

1/ 文献调研

2/ 文献复现

3/ 短篇小说数据集

4/ 程序项目结构

5/ 模型训练

6/ 分析与总结

■ 情感风格迁移 ■ ■

情感是一种比较容易定义的风格，也是目前研究得较多的方向
以下列举了部分典型文献使用的思路

- 定位并直接删除情感词^[1]
- 将情感词替换为具有相反情感的词^[2]
- 预先指定语句模板，再进行生成^[3]
- 逐个标记单词并进行删除/插入/替换操作^[4] (Point-Then-Operate)
- 使用VAE进行情感和内容的分离，再进行组合^[5]
- 使用Back-translation来对比文本语义^[6]
- 双向强化学习方法^[7]

□ 方法限制较大

□ 对数据要求过高

■ 风格迁移问题 ■ ■

根据对现有文献的分析，我们总结了风格迁移问题的几个关键点

◆ 何为写作风格

- 文本中对句式的偏好、用词的倾向（例如正式/非正式文字）
- 类似于图像中的纹理、配色等特征，而非图像中的内容和语义

◆ 如何迁移

- 迁移后需要符合另一种写作风格的特征
- 迁移后需要保证语句通顺
- 迁移前后语义不能发生太大变化
- 迁移前后要保证都是完整的语句，而迁移过程不限



清华大学
Tsinghua University

CONTENTS

目录

1/ 文献调研

2/ 文献复现

3/ 短篇小说数据集

4/ 程序项目结构

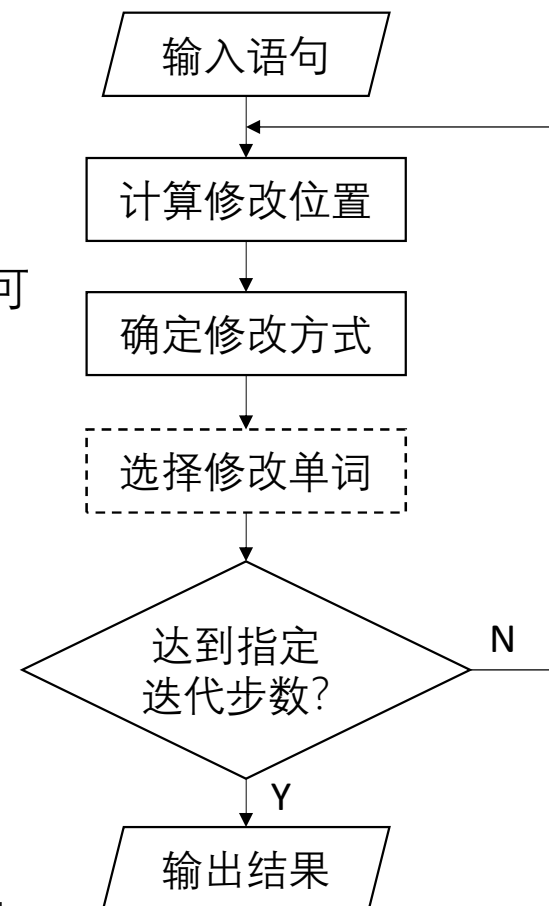
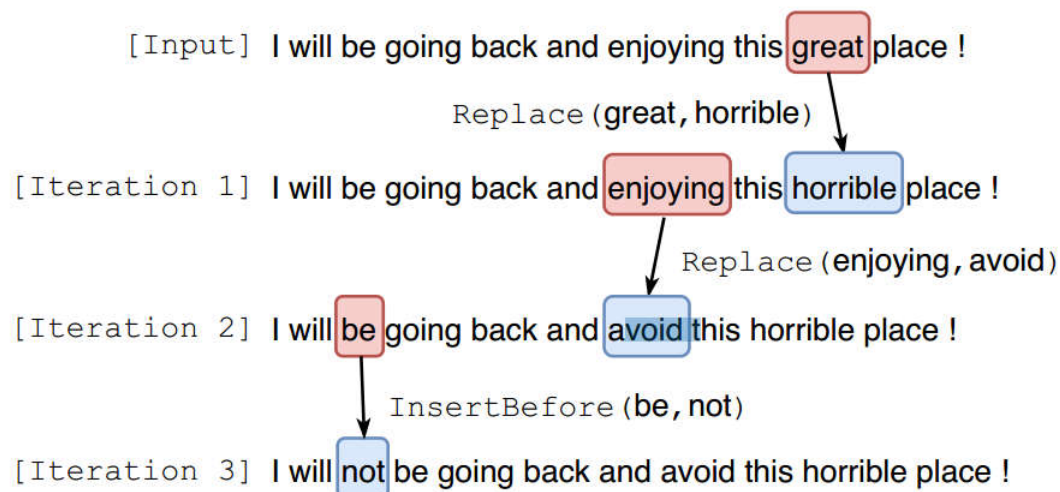
5/ 模型训练

6/ 分析与总结

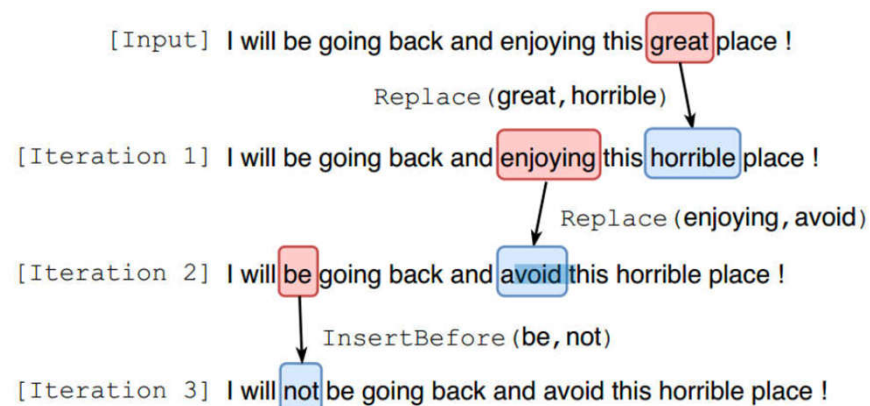
■ 算法思路

□ Point-Then-Operate

- Point: 指出需要修改的位置
- Operate: 对单个位置、单个单词进行修改
- 修改方式包括删除、插入、替换和不变
- 因为允许选择不修改，迭代次数可固定，足够多即可
- 贪心策略，每一步迭代都使损失最小/回报最大



文献复现



分级强化学习

- 定位修改位置——Pointer（顶层）
- 确定修改方法——Operator（底层）
- 迭代修改

■ 文献复现

回报函数设计

- 语言模型回报——语句通顺程度
- 风格极性回报——语句风格
- 语句内容损失——语句语义

■ 文献复现

神经网络架构

- 语言模型——使用Encoder、RNN和Decoder判断语句出现的概率，判断语句通顺度
- 风格分类器——基于TextCNN的二分类器
- Attention 分类器——判断修改位置
- InsFront 模型——判断单词前插入位置
- InsBack 模型——判断单词后插入位置
- Replace 模型——判断替换单词位置

■ 文献复现

训练顺序

- 语言模型和分类器模型
- Pointer模型和Attention模型
- 分级强化学习

■ 文献复现

训练结果

在风格迁移的准确率和语句通顺程度上较原文差，推测主要原因是训练次数不够多

遇到的问题

Pytorch环境配置问题



清华大学
Tsinghua University

CONTENTS

目录

1/ 文献调研

2/ 文献复现

3/ 短篇小说数据集

4/ 程序项目结构

5/ 模型训练

6/ 分析与总结

■ 短篇小说数据集

□ 选取的作品

- 马克·吐温《傻瓜威尔逊》（The Tragedy of Pudd'nhead Wilson）前两章内容
- 爱伦坡《梅氏男爵》（Metzengerstein）全部内容
- 每部作品选取了20KB左右的文本

□ 数据集特点

- 实际的文学作品中，句式变化丰富，含有大量缩写、特殊单词等，处理麻烦
- 长句很多，需要选取合适的最大语句长度值并进行适当处理
- 有很多单词出现频率很低，需要在词典很大或文本中出现大量<unk>之间权衡
- 没有成对数据，各语句之间语义相差基本都很大，只有风格标签
- 网上不一定能直接下载到格式比较清晰的文本，需要大量手动处理的工作

数据集格式

词典

- 特殊字符: `<pad>`, `<go>`, `<eos>`, `<unk>`
- `word2id`: 单词编号
- `id2word`: 编号到单词的映射

数据样本

- 填充后的文本: 在语句后添加填充符`<pad>`直至最大长度
- 文本编码: 数据集中只储存各单词的编码, 包括填充符, 维度为 $[N, max_len]$
- 语句有效长度: 不含填充符时语句的长度
- 填充符蒙版`null_mask`: 指示填充符位置的蒙版
- 风格标签: 取0或者1

数据示例

- Pudd'nhead wins his name.
- Pudd'nhead wins his name `<pad>` `<pad>` `<pad>` `<pad>` `<pad>` `<pad>` `<pad>`
- `[4, 6, 7, 10, 3, 3, 3, 3, 3, 3]`
- `Null_mask`: `[0, 0, 0, 0, 1, 1, 1, 1, 1, 1]`



清华大学
Tsinghua University

CONTENTS

目录

1/ 文献调研

2/ 文献复现

3/ 短篇小说数据集

4/ 程序项目结构

5/ 模型训练

6/ 分析与总结



■ 程序模块

□ 风格迁移的要求

- 迁移后需要符合另一种写作风格的特征
- 迁移后需要保证语句通顺
- 迁移前后语义不能发生太大变化

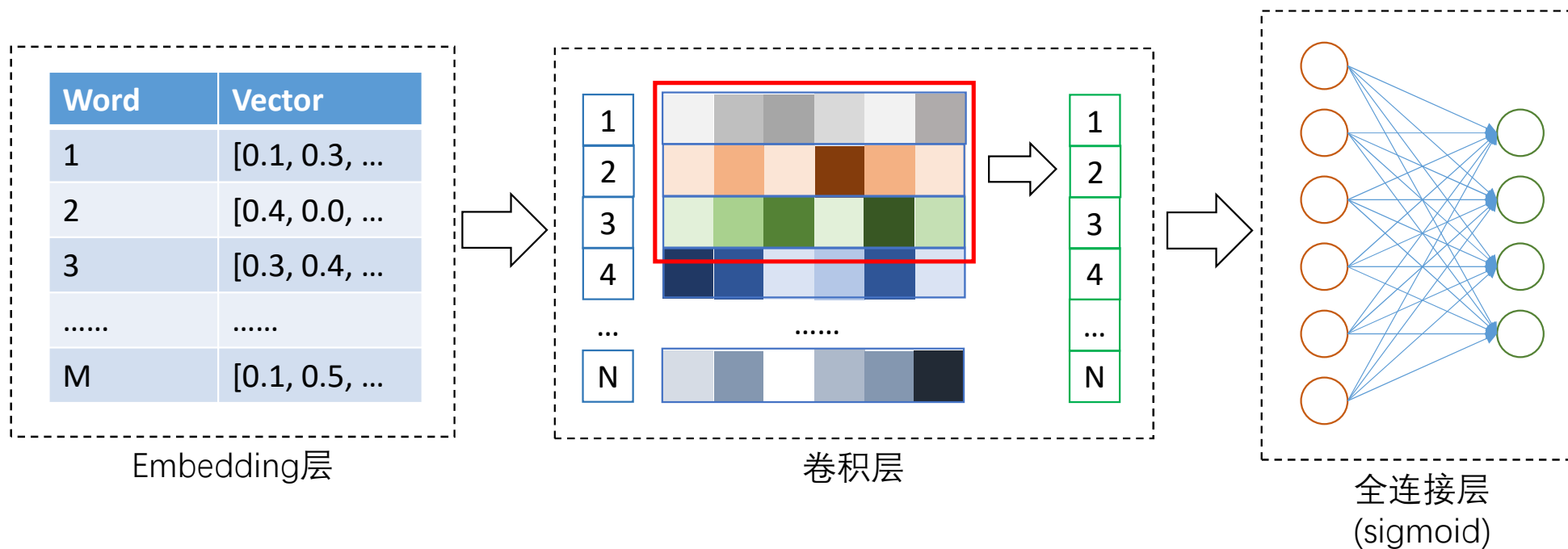
□ 相应的程序模块

- 语句风格分类器（二分类）
- Attention-Based语句风格分类器（Pointer）
- 语言模型计算语句概率（通顺性）
- Seq2Seq自编码器（语义对比）
- Operator所需的单词选择模型（本质上还是分类器）

■ 风格分类器

□ TextCNN

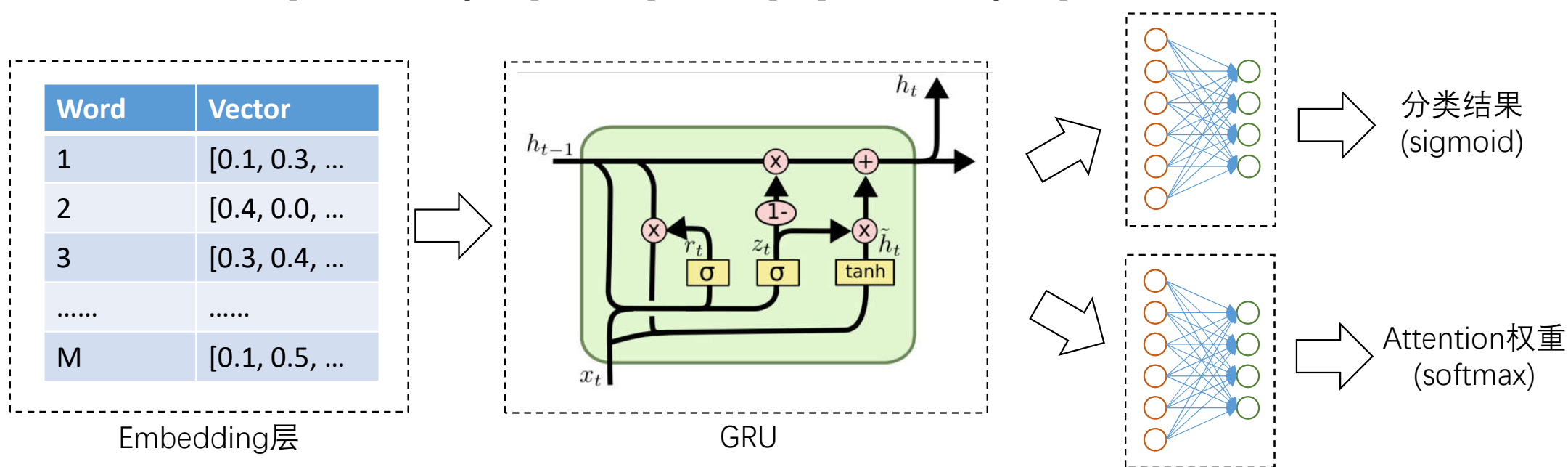
- 使用与Embedding层同宽的卷积核提取特征
- 不使用循环神经网络
- 输入 $[batch\ n, seq\ len]$, 输出 $[batch\ n]$, 值在0 ~ 1之间



Pointer模块

Attention Classifier

- 中间使用多层循环神经网络
- 输出用于分类，隐藏状态用于计算Attention权重
- 可使用分类任务预训练
- 输入[batch n, seq len], 输出[batch n]和[batch n, seq len]



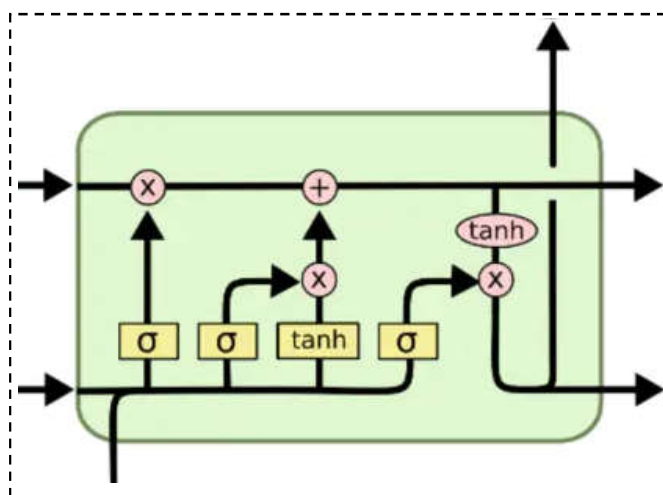
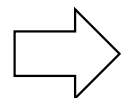
语言模型

Language Model

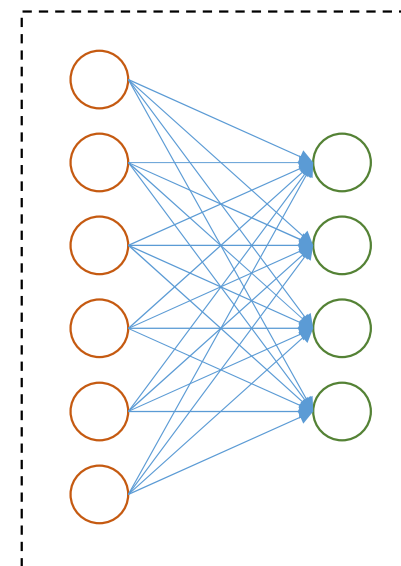
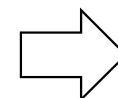
- 输入语句内容，对下一个单词进行预测，输出的是各单词的出现概率
- 全部单词出现概率相乘即为语句出现概率
- 输入[*batch n*, *seq len*], 输出[*batch n*, *seq len*, *vocab size*]

Word	Vector
1	[0.1, 0.3, ...
2	[0.4, 0.0, ...
3	[0.3, 0.4, ...
.....
M	[0.1, 0.5, ...

Embedding层



LSTM

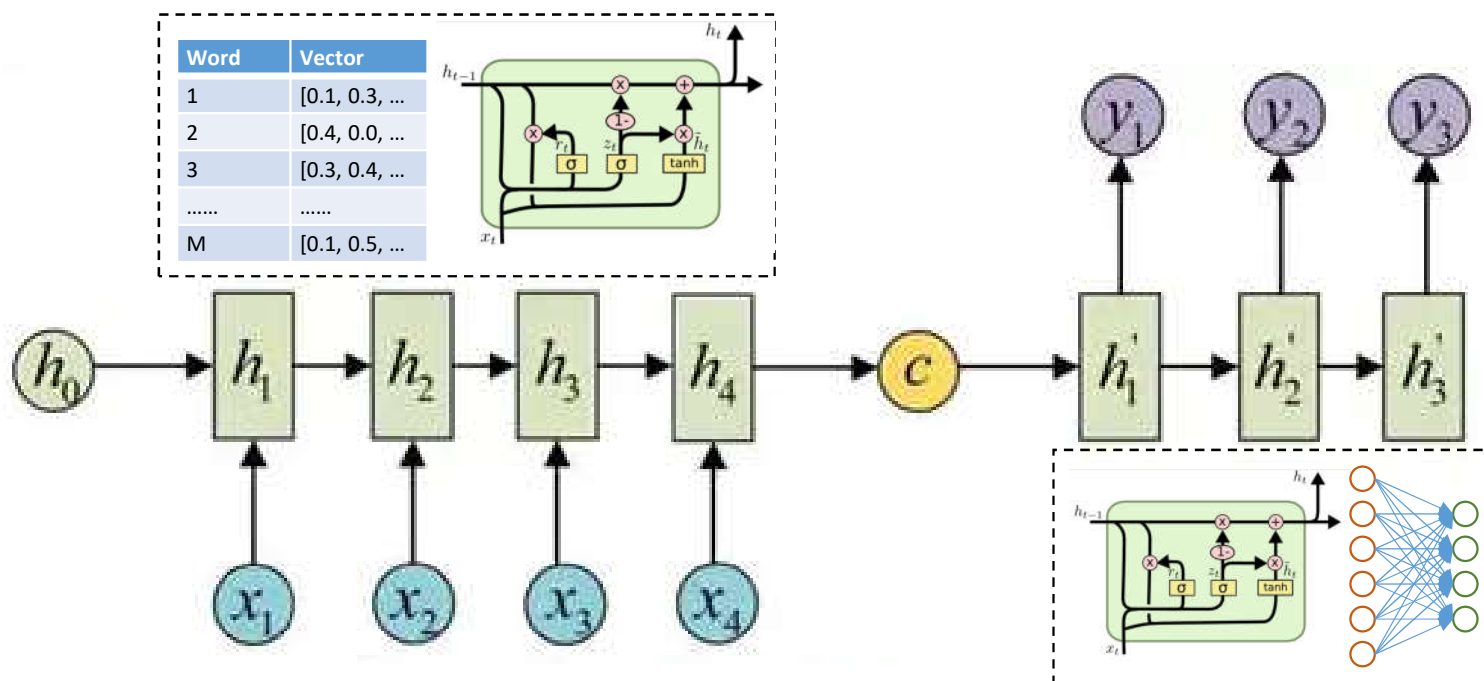


全连接层
(sigmoid)

■ 自编码器模型

□ Seq2Seq AutoEncoder

- 非监督，只使用语句数据进行训练
- 最终只使用编码器，用隐藏状态表示语义
- 用隐藏状态的MSE表示两语句语义的差别

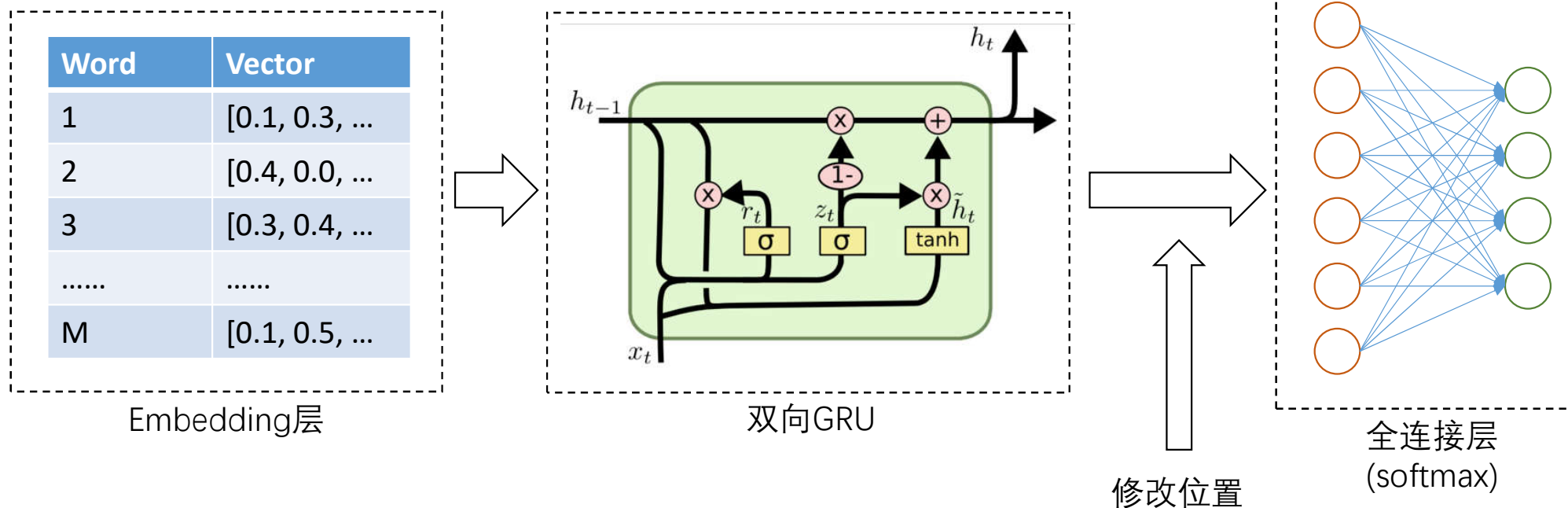




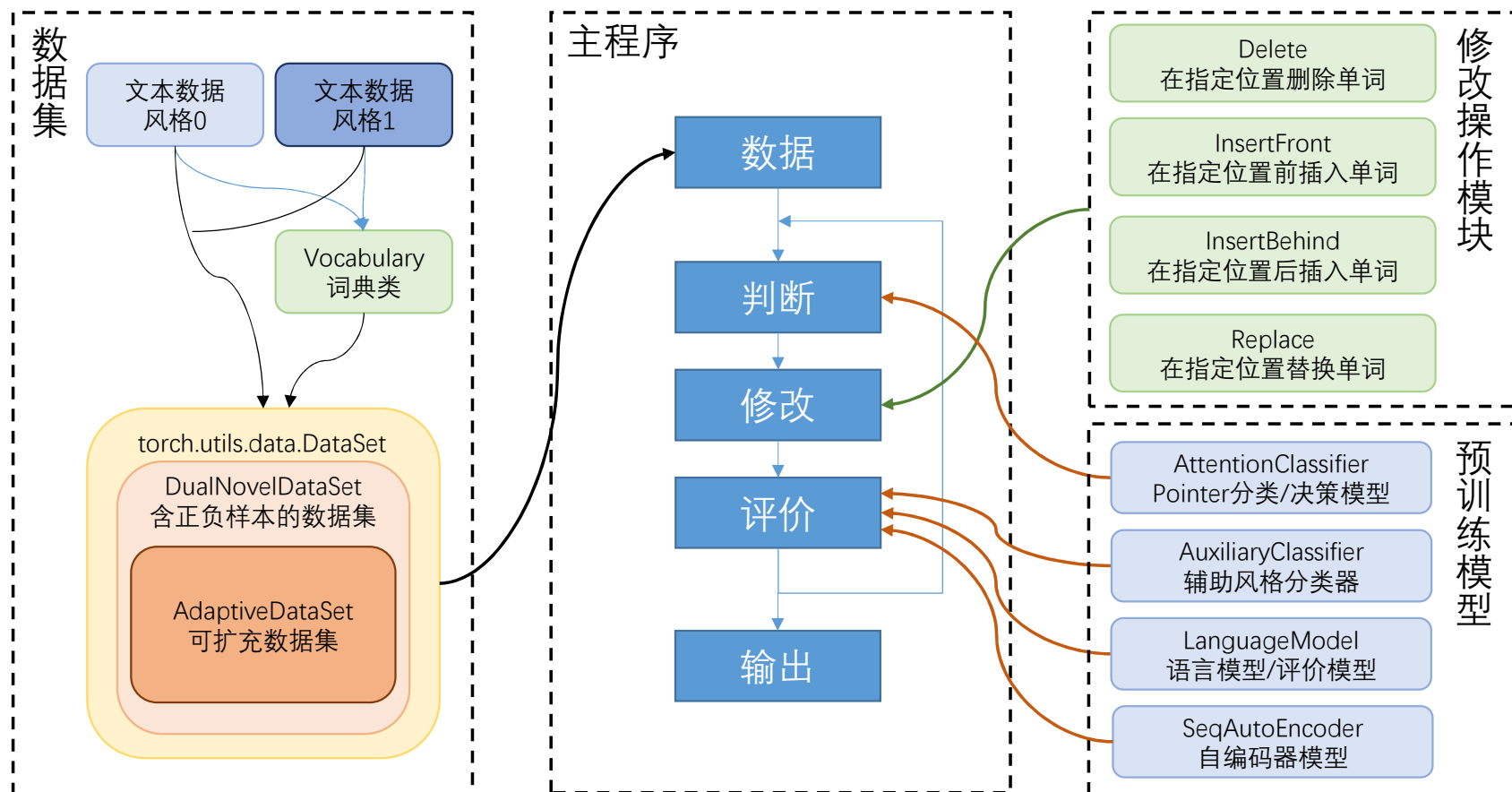
Operator模型

□ RNN-based Classifier

- 使用双向循环神经网络计算每个位置选择各单词的概率
- 根据事先确定的待修改位置，确定每一句修改时各单词选取的概率
- 输入[batch n , seq len], 输出[batch n , vocab size]



■ 项目结构 & 程序流程





■ (加权)损失函数

□ 风格变化损失

- 分类器输出的0 ~ 1值代表风格的极性
- 风格回报和损失

$$Reward_{style} = |Class_0 - Class_{edited}| \in [0,1]$$

$$Loss_{style} = 1 - Reward_{style}$$

□ 通顺程度损失

- 语言模型输出的0 ~ 1概率值代表语句的通顺程度
- 通顺程度损失

$$Loss_{fluency} = 1 - Prob_{edited}^{(LM)}$$

□ 语义损失

- Seq2Seq自编码器输出编码结果可以表示语义
- 编码结果之差可以表示语义变化
- 语义损失

$$Loss_{semantic} = MSE(Encoded_0, Encoded_{edited})$$

$$Total_{Loss} = \alpha \cdot Loss_{style} + \beta \cdot Loss_{fluency} + \gamma \cdot Loss_{semantic}$$

■ 可扩充数据集

□ 迭代更新存在的问题

- 原论文程序的训练过程中，每次都取没有修改过的语句，只进行一次训练，而且方法选择有限
- 参考强化学习中经验回放的机制，将修改过的语句重新加入数据集

□ 改进

- 每次训练时，对取出的数据将所有的修改方法都尝试一遍
- 当风格回报 $Reward_{style}$ 超过阈值时，就将修改过后的语句重新加入数据集
- 训练超过一定轮数后就不再添加新数据，防止数据集过大
- 实际训练过程中，可以将数据集扩展至原来的15~20倍左右，有效解决了原始数据集过小带来的问题



清华大学
Tsinghua University

CONTENTS

目录

1/ 文献调研

2/ 文献复现

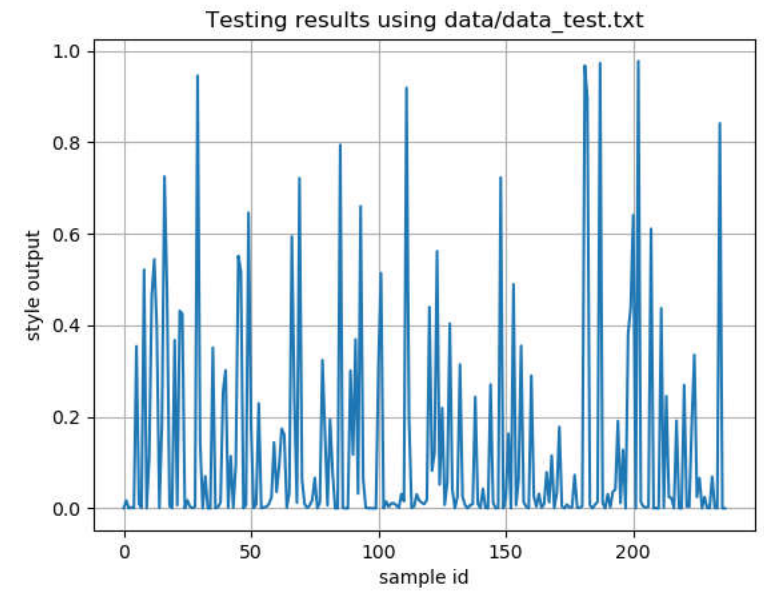
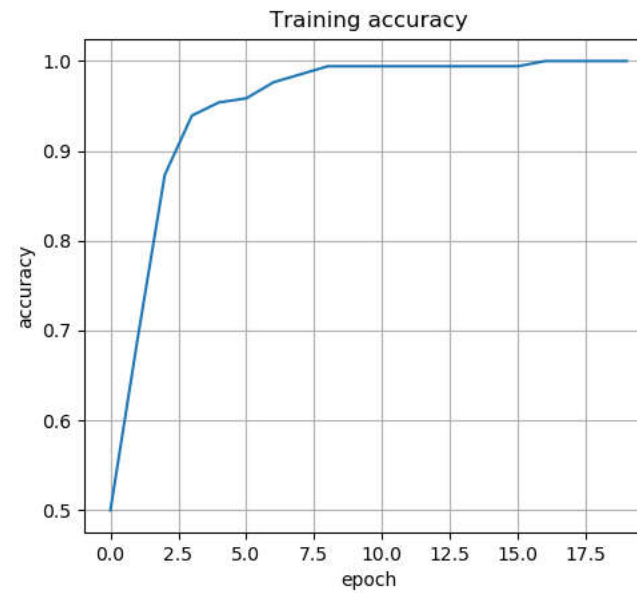
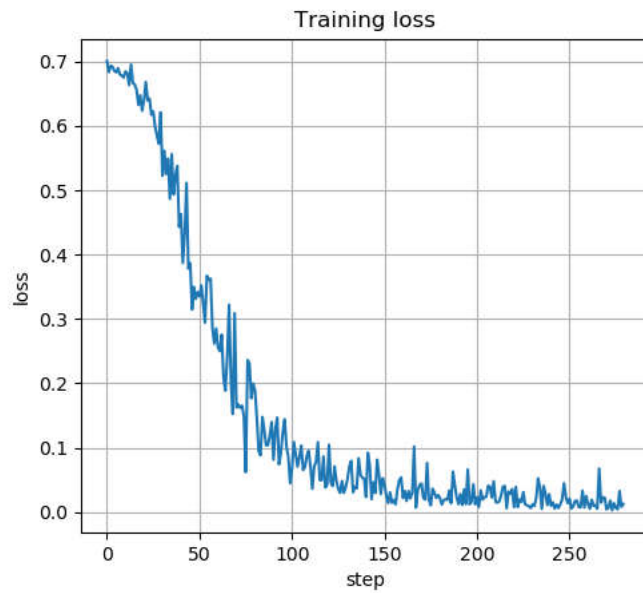
3/ 短篇小说数据集

4/ 程序项目结构

5/ 模型训练

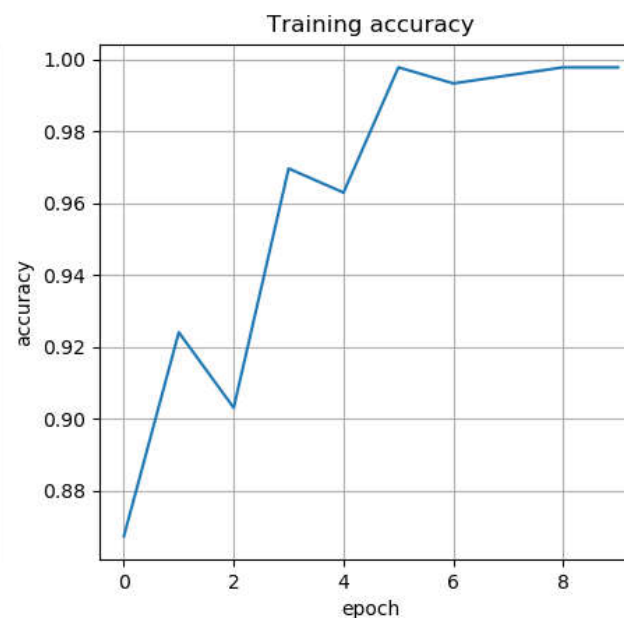
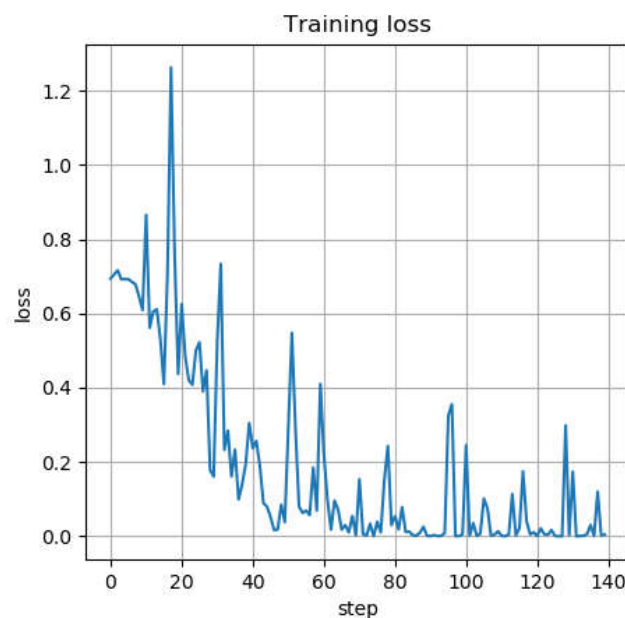
6/ 分析与总结

■ 风格分类器





■ Pointer模块

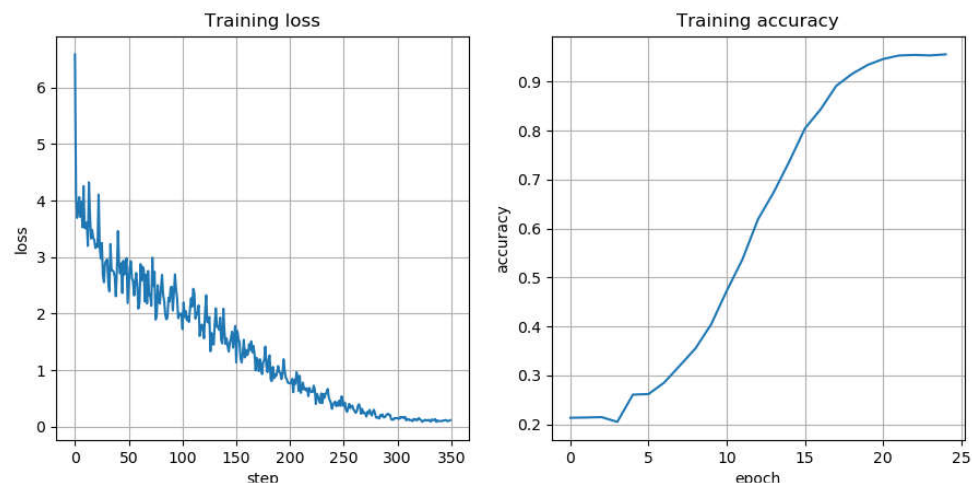


pudd'nhead wins his name
horror and fatality have been stalking abroad in all ages



风格对数据量有要求

语言模型

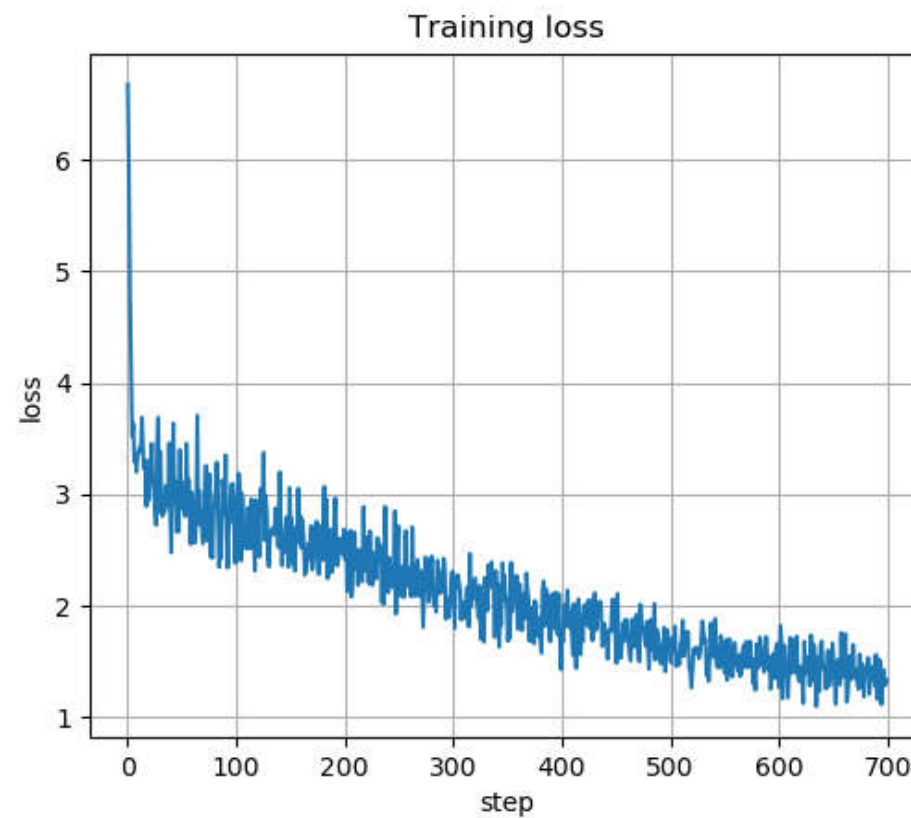


概率	语句
0.740	pudd'nhead <unk> his name
0.782	tell the truth or <unk> but get the <unk>
0.003	the <unk> of this <unk> is the town of dawson's landing on the missouri <unk> of the mississippi half a day's <unk> per <unk> <unk> st
0.301	a lofty name shall have a fearful fall when as the rider over his horse the <unk> of metzengerstein shall triumph over the <unk> of berlifitzing
0.814	fire and <unk> might <unk> <unk> than a berlifitzing <unk> the hand of a metzengerstein
0.642	when there was <unk> on the <unk> outside of the pots and boxes for a cat the cat was there in <unk> <unk> stretched at full length <unk> and <unk>
0.185	then that house was <unk> and its <unk> and <unk> were made <unk> to the world by this <unk> whose <unk> is <unk>

■ 自编码器模型



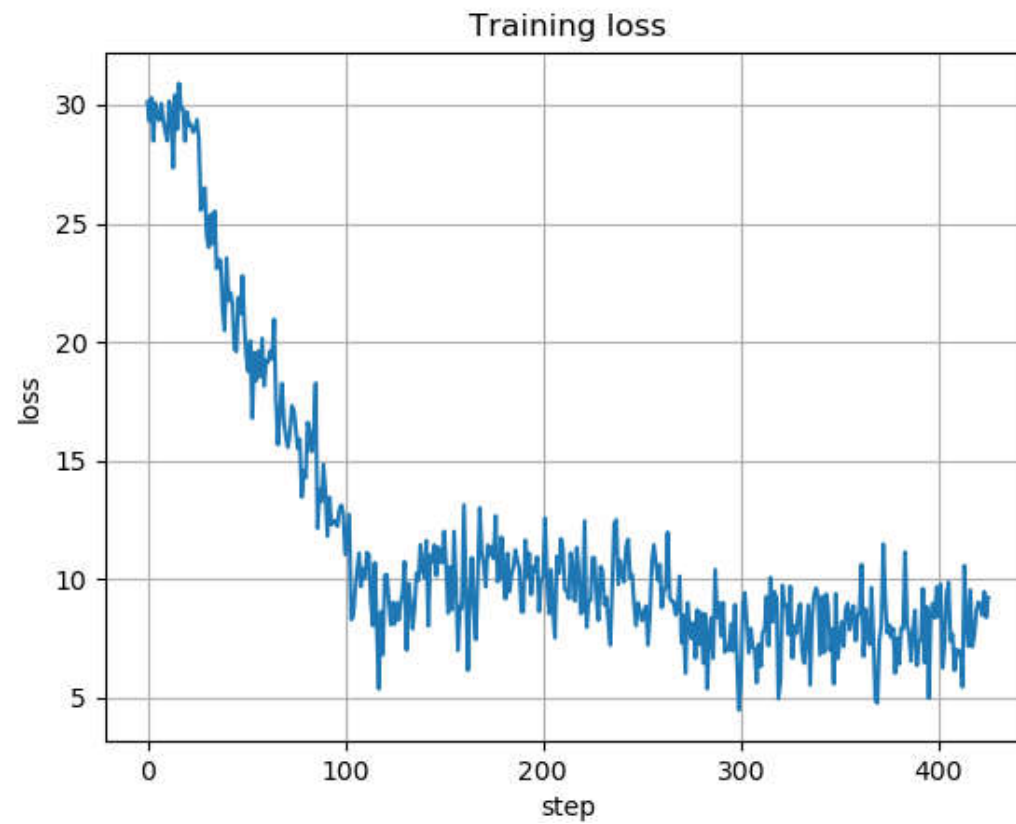
清华大学
Tsinghua University



■ PTO模型



清华大学
Tsinghua University





清华大学
Tsinghua University

CONTENTS

目录

1/ 文献调研

2/ 文献复现

3/ 短篇小说数据集

4/ 程序项目结构

5/ 模型训练

6/ 分析与总结

■ 迁移结果

原有风格	迁移结果	
0	迁移前	driscoll had said it and always in the same <unk> tone
	迁移后	noble noble driscoll had said it and always in the same <unk> tone
0	迁移前	he was a fairly humane man toward slaves and other <unk> he was an <unk> humane man toward the <unk> of his own race
	迁移后	he was only humane man toward slaves and other <unk> he was an <unk> humane man toward the <unk> of his own race
1	迁移前	he is your own property <unk>
	迁移后	he is your own noble noble property <unk>
1	迁移前	to be sure the words themselves had little or no meaning
	迁移后	the only themselves had little or no meaning

■ 总结

□ 完成的内容

- 对现有文献中Point-Then-Operate方法的复现和改进
- 将文献的开源程序进行了深度重构，对程序、模型与项目结构进行了精简
- 基于自编码器模型，实现了语义损失项的计算，并整理了损失函数的结构
- 设计并应用了可扩展数据集，在小数据集上完成了模型的训练

□ 可进一步改进的内容

- 分类模型中，需要保证模型提取的是风格信息，而不是与文本有关的其他特征（比如文本中特殊的人名、地名等）
- PTO方法容易实现但效果局限性比较大，例如不能实现句式的修改
- 使用额外的数据进行预训练，避免训练数据集过小引起的问题
- 可以用于生成成对的数据，进一步训练其他模型

■ 参考文献

□ 方法调研过程中所参考的文献

- [1] J. Xu, X. Sun, Q. Zeng, X. Ren, X. Zhang, H. Wang and W. Li, “Unpaired Sentiment-to-Sentiment Translation: A Cycled Reinforcement Learning Approach,” Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, 2018.
- [2] J. Li, R. Jia, H. He, P. Liang, “Delete, Retrieve, Generate: A Simple Approach to Sentiment and Style Transfer,” arXiv eprints: 1804.06437v1, 2018.
- [3] K. Guu, T. B. Hashimoto, Y. Oren, P. Liang, “Generating Sentences by Editing Prototypes,” arXiv eprints: 1709.08878v2, 2018.
- [4] C. Wu, X. Ren, F. Luo, X. Sun, “A Hierarchical Reinforced Sequence Operation Method for Unsupervised Text Style Transfer,” Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019.
- [5] T. Shen, T. Lei, R. Barzilay, T. Jaakkola, “Style Transfer from Non-Parallel Text by Cross-Alignment,” arXiv eprints: 1705.09655, 2017.
- [6] S. Prabhumoye, Y. Ysvetkov, R. Salakhutdinov, A. W. Black, “Style Transfer Through Back-translation,” arXiv eprints: 1804.09000v3, 2018.
- [7] F. Luo, P. Li, J. Zhou, P. Yang, B. Chang, X. Sun, Z. Sui, “A Dual Reinforcement Learning Framework for Unsupervised Text Style Transfer,” Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI-19), 2019.

■ 参考文献

□ 与语义相似度研究相关的文献

- [8] Kusner M, Sun Y, Kolkin N, et al. From word embeddings to document distances[C]//International conference on machine learning. PMLR, 2015: 957-966.
- [9] Huang P S, He X, Gao J, et al. Learning deep structured semantic models for web search using clickthrough data[C]//Proceedings of the 22nd ACM international conference on Information & Knowledge Management. 2013: 2333-2338.
- [10] Shen Y, He X, Gao J, et al. A latent semantic model with convolutional-pooling structure for information retrieval[C]//Proceedings of the 23rd ACM international conference on conference on information and knowledge management. 2014: 101-110.
- [11] Hu B, Lu Z, Li H, et al. Convolutional neural network architectures for matching natural language sentences[C]//Advances in neural information processing systems. 2014: 2042-2050.
- [12] Palangi H, Deng L, Shen Y, et al. Deep sentence embedding using the long short term memory network: Analysis and application to information retrieval. CoRR abs/1502.06922 (2015) [J].
- [13] Lu Z, Li H. A deep architecture for matching short texts[J]. Advances in neural information processing systems, 2013, 26: 1367-1375.
- [14] Hu B, Lu Z, Li H, et al. Convolutional neural network architectures for matching natural language sentences[C]//Advances in neural information processing systems. 2014: 2042-2050.
- [15] Pang L, Lan Y, Guo J, et al. Text matching as image recognition[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2016, 30(1).
- [16] Wan S, Lan Y, Xu J, et al. Match-srnn: Modeling the recursive matching structure with spatial rnn[J]. arXiv preprint arXiv:1604.04378, 2016.



清华大学
Tsinghua University

THE END
THANK YOU

