123123

测试

分 类 号 学 号 M201873134

学校代码 10487 密 级



硕士学位论文

**基于依存句法分析和语言模型的细粒度文本情感转换**

|  |  |
| --- | --- |
| **学位申请人：** | **肖露露** |
| **学科专业：** | **计算机应用技术** |
| **指导教师：** | **李瑞轩 教授** |
| **答辩日期：** | **2021年5月25日** |
|  |  |

**A Dissertation Submitted in Partial Fulfillment of the Requirements for the Master Degree in Engineering**

Fine-grained Text Sentiment Transfer via Dependency Parsing and Language Model

|  |  |
| --- | --- |
| **Candidate:** | XIAO Lulu |
| **Major :** | **Computer Applied Technology** |
| **Supervisor:** | Professor LI Ruixuan |
|  |  |

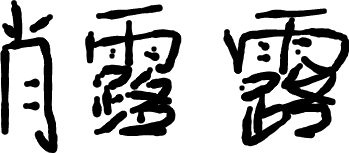
Huazhong University of Science and Technology

Wuhan 430074, P.R.China

May, 2021

**独创性声明**

本人声明所呈交的学位论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除文中已经标明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的研究成果。对本文的研究做出贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名：

日期：2021年 05月 28日

**学位论文版权使用授权书**

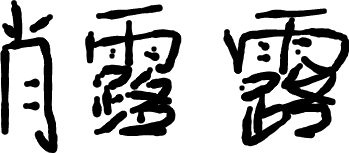
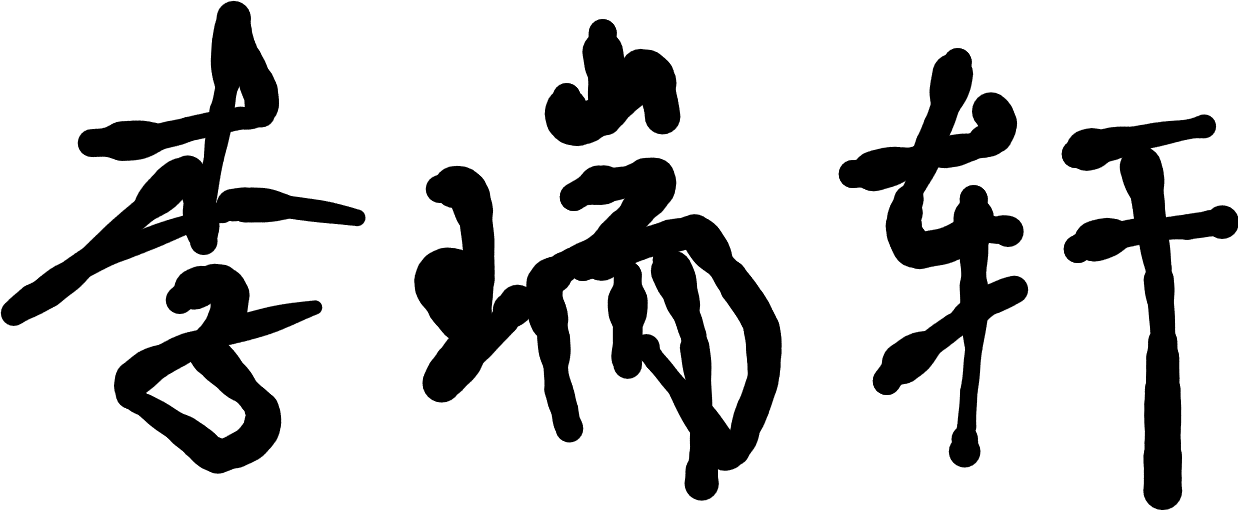
本学位论文作者完全了解学校有关保留、使用学位论文的规定，即：学校有权保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权华中科技大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

保密□， 在 年解密后适用本授权书。

本论文属于

不保密☑。

（请在以上方框内打“√”）

学位论文作者签名： 指导教师签名：

日期：2021年 05月 28日 日期：2021年05月28日

# 摘要

文本风格转换（Text Style Transfer）旨在对文本进行改写以满足目标风格，同时保留其原始语义内容。文本情感转换（Text Sentiment Transfer）是文本风格转换的重点研究内容，传统的文本情感转换主要集中在积极（positive）和消极（negative）这两个情感极性的转换上，转换粒度较为粗糙，难以应用在复杂的情感转换任务上。并且传统的文本情感转换方法倾向于将文本的语义内容和情感内容分开，然而这两个因素在一定程度上很难完全分开，并且这对于文本改写也是不必要的。

细粒度文本情感转换（Fine-grained Text Sentiment Transfer）将情感转换扩展到更普遍的场景中，在给定情感强度值为1到5的情况下修改文本。强度1到5的情感值分别对应于强消极、弱消极、中性（neutral）、弱积极和强积极五种情感。为了更好地实现文本细粒度情感转换，针对现有文本情感转换方法的不足，提出了基于依存句法分析的细粒度情感转换模型（Fine-grained Text Sentiment Transfer Model Base On Dependency Parsing，FGSTDP）和基于语言模型的细粒度情感转换模型（FGSTDP + LM）。这两种模型在原始文本上直接进行修改，并将情感强度值作为解码器的额外输入，以实现细粒度的情感控制。由于观察到文本中情感词（例如，“delicious”）与特定的非情感上下文词（例如，“food”）具有紧密关系，因此可以使用依存句法分析来捕获句子中与情感词有特定依存关系的上下文词汇，以此构造伪并行句子对。利用伪平行句子与生成句子之间的差异以及其他约束条件来指导模型精确修改文本并保留原始语义内容，以减轻模型的负担。FGSTDP+LM在FGSTDP的基础上加入语言模型，以提高生成文本的句子流畅度。

通过在真实数据集Yelp上的实验表明，FGSTDP和FGSTDP+LM模型在四个评价指标MAE、BLEU、Edit Distance和Fluency上均优于现有模型，这验证了模型在内容完整性和情感转换准确性上的有效性。

**关键词：**文本风格转换；细粒度情感转换；依存句法分析；语言模型

# Abstract

Text style transfer aims to rewrite a text to require the given style while keeping its original semantic content unchanged. Text sentiment transfer is an important research in text style transfer. Traditional text sentiment transfer mainly focuses on the two sentiment polarities of positive and negative. The conversion granularity is relatively rough. It is difficult to apply to complex sentiment transfer tasks. And traditional text sentiment transfer methods try to separate the semantic content and sentiment content of the text. However, it is difficult to completely separate these two factors, and it is also not necessary.

Fine-grained text sentiment transfer extends to more general scenes, and modifies the text under the given sentiment intensity value ranging from 1 to 5. The value of intensity 1 to 5 respectively corresponds to strong negative, weak negative, neutral, weak positive, and strong positive. In order to achieve fine-grained text sentiment transfer and solve the shortcomings of existing methods, we propose fine-grained sentiment transfer model based on dependency parsing(FGSTDP) and fine-grained sentiment transfer model based on language model(FGSTDP+LM). The two models directly modifie the original text, and uses the sentiment value as the extra input of the decoder to achieve fine-grained sentiment control. Since it is observed that sentiment words (e.g., "delicious") in the text have a close relationship with non-sentiment context words (e.g., "food"). Then dependency parsing is used to capture the words that have the certain dependency with the sentiment words to constructs pseudo-parallel sentences. The difference between pseudo-parallel sentences and generated sentences and other constraints are used to guide the model to accurately modify text and retain the original semantic. FGSTDP+LM adds a language model on the basis of FGSTDP to improve the fluency of generated text.

Experiments on the dataset Yelp show that FGSTDP and FGSTDP+LM models outperform the existing methods in four evaluation metrics MAE, BLEU, Fluency and Edit Distance, which shows the effectiveness of the models in content integrity and accuracy of sentiment transfer.

**Key words:** text style transfer, fine-grained sentiment transfer, dependency parsing,

language model

# 目 录

[摘要 I](#_Toc71128357)

[Abstract II](#_Toc71128358)

1 绪论

[1.1 研究背景 （1](#_Toc71128361)）

[1.2 研究目的和意义 （2](#_Toc71128362)）

[1.3 文本情感转换国内外研究现状 （3](#_Toc71128363)）

[1.4 论文的主要研究内容 （7](#_Toc71128364)）

[1.5 论文的组织结构 （8](#_Toc71128365)）

[2 相关理论基础](#_Toc71128366)

[2.1 依存句法分析 （10](#_Toc71128367)）

[2.2 循环神经网络 （12](#_Toc71128368)）

[2.3 编码器-解码器结构与注意力机制 （15](#_Toc71128369)）

[2.4 本章小结 （18](#_Toc71128370)）

[3 基于依存句法分析的细粒度文本情感转换模型](#_Toc71128371)

[3.1 模型基本定义和总体结构 （19](#_Toc71128372)）

[3.2 情感内容提取 （19](#_Toc71128373)）

[3.3 伪平行句子对构造 （21](#_Toc71128374)）

[3.4 基于编码器的多分类情感模型 （24](#_Toc71128375)）

[3.5 基于依存句法分析的情感转换模型结构 （26](#_Toc71128376)）

[3.6 基于依存句法分析的情感转换模型训练 （27](#_Toc71128377)）

[3.7本章小结 （29](#_Toc71128378)）

[4 基于语言模型的细粒度文本情感转换模型](#_Toc71128379)

[4.1 模型基本定义和总体结构 （30](#_Toc71128380)）

[4.2 训练前的准备工作 （30](#_Toc71128381)）

[4.3 基于双向GRU的语言模型 （31](#_Toc71128382)）

[4.4 基于语言模型的情感转换模型结构 （33](#_Toc71128383)）

[4.5 基于语言模型的情感转换模型训练 （34](#_Toc71128384)）

[4.6 本章小结 （35](#_Toc71128385)）

[5 实验与分析](#_Toc71128386)

[5.1 实验环境及数据集 （36](#_Toc71128387)）

[5.2 实验评价指标 （37](#_Toc71128388)）

[5.3 实验对比算法 （39](#_Toc71128389)）

[5.4 实验设置与方案设计 （40](#_Toc71128390)）

[5.5 实验结果及分析 （41](#_Toc71128391)）

[5.6 本章小结 （47](#_Toc71128392)）

[6 总结与展望](#_Toc71128393)

[6.1 论文总结 （49](#_Toc71128394)）

[6.2 论文展望 （50](#_Toc71128395)）

[致谢 （51](#_Toc71128396)）

[参考文献 （53](#_Toc71128397)）

[附录1 攻读硕士学位期间参与的科研项目 （60](#_Toc71128398)）

[附录2 攻读硕士学位期间发表的论文 （61](#_Toc71128399)）

# [1 绪论](#_Toc283328270)

本章首先介绍文本风格转换和文本细粒度情感转换的研究背景，然后引出本文研究课题的目的和意义。接着介绍文本风格转换和文本细粒度情感转换的现有研究方法，总结这些方法的缺陷与不足。然后介绍本文的主要研究内容，最后描述本文的组织架构。

## 1.1 研究背景

1 一级标题

1.1二级标题

1.1.1三级标题

风格转换的研究源于图像领域[1][2][3]，经过学者们长期的研究，风格转换在图像领域取得了十分不错的成果，并且技术趋于成熟。后来风格转换逐渐应用于文本领域，吸引了无数研究者的目光，越来越多的技术应运而生。文本风格转换是一项在给定原始文本和特定的风格（如书写风格、情感、正式化程度、时态等）下，对原始文本的风格进行转换的任务。文本风格转换可用于自动改变文本的文风（比如改写成风趣幽默、清新简洁等风格）、社交平台评论文本情感改写、协助论文邮件创作等文本编辑任务上[4][5]。由于文本的离散性，图像方面用于风格转换的技术无法适用于文本，于是学者们提出使用编码器-解码器框架及其变形[6][7]来解决该问题。

文本情感转换是文本风格转换的重点研究内容，其目的是在保证原始文本语义内容完整性的同时将文本的情感（消极、积极）转换为相反情感，即将消极性文本转换成积极性文本，反之亦然。文本情感转换吸引了无数研究人员的注意力，产生了大量有意义有价值的研究工作。现有的文本情感转换方法主要集中于消极、积极二元情感转换，在转换粒度上较为粗糙，难以应用在复杂多样的文本情感转换任务上。并且现有的方法大多倾向于将文本的语义内容和情感内容分开，然后再在与情感无关的内容上进行修改。然而语义内容和情感内容很难完全分开，因为情感内容也是语义内容的一部分。因此在分开之后的语义内容上进行修改会使得生成的文本丢失一部分语义信息，尤其对于表达比较含蓄的文本来说会损失重要的语义信息。并且大部分现有的工作会根据情感的种类来构建多个解码器，不同的解码器生成不同情感的文本，这种方法相对没有那么灵活，对于细粒度情感转换十分不友好。

细粒度文本情感转换在二元情感极性转换上进行扩展，可以适应更普遍的场景。细粒度情感转换根据给定范围为1到5的情感强度值修改句子，这里的情感强度值从之前的二元情感扩展为更细粒度的多元情感。1到5的数值分别代表强消极、弱消极、中性、弱积极和强积极五种情感强度，因此细粒度情感转换不仅可以将文本原始情感转换为相反情感，还可以转换为不同情感强度甚至是中性的句子。比如给定情感强度值为4的原始文本“the food was totally fine”，细粒度情感转换可以将其转换为情感强度值为3的中性文本“the food was enough”，也可以转换为表达更强烈积极情绪的情感值为5的文本“the food was totally wonderful”。或者转换成情感强度值为2的表达弱消极情绪的文本“the food was forgettable”，以及转换成表达强消极情绪的情感强度值为1的文本“the food was totally terrible”。细粒度文本情感转换任务旨在修改原始文本，在保持原始语义内容的同时满足目标情感强度值。此任务具有一定的挑战性，并且现有的方法存在着不足之处。首先，生成句子时很难实现对情感强度的细粒度控制；其次，由于没有真实的并行数据，无法使用有监督的方式来训练模型；并且现有的方法试图将句子分解为语义内容部分和情感部分，但是由于这两部分以复杂的方式混合在一起，很难将它们完全分开，这通常导致原始句子的语义与生成句子的语义相差甚远。

## 1.2 研究目的和意义

本课题研究细粒度文本情感转换，与文本的二元极性情感转换不同的是，细粒度情感转换将文本表达的情感从消极到积极分为多个过渡级别。本课题旨在实现更细粒度的文本情感转换，以更好地满足实际应用中的情感转换需求。

由于缺乏真实的并行数据，即原始文本没有包含对应的目标情感强度值的真实改写数据，现有的方法大多采用无监督的方式训练模型。为了减轻模型的负担，本文引入了伪并行句子对，以指导模型在给定情感强度值上生成符合要求的文本。伪并行句子对是具有相似语义内容但情感值不同的句子。由于观察到情感词“delicious”更适合修饰“food”而不是“staff”，并且不同的情感词表达了不同强度的情感（例如，“delicious”比“ok”具有更强的积极情绪，“terrible”表达的消极情绪比“so-so”强）。依存句法分析可以捕捉句子中词汇间的依存关系，因此可以利用依存句法分析找到与情感词具有特定依存关系的非情感上下文词。利用这种关系，辅助伪并行句子的生成。通过计分器函数评估所有描述同一上下文的情感词，选择最佳的情感词来代替原始文本中的情感词，以此获得与目标情感值对应的伪平行句子。然后利用目标情感值对应的伪平行句子与原始句子的差异来辅助模型修改文本。在此基础上，为了解决生成文本语言流畅性差的问题，引入了语言模型来提高文本的流畅性。因此本课题的研究目的为使用编码器-解码器序列到序列模型，在情感强度的控制下生成具有原始语义内容并且满足目标情感强度值的新文本。为了减轻生成模型的负担，使用依存句法分析找到与情感内容具有特定依存关系的上下文，以此辅助构造伪平行句子对，提高生成模型细粒度情感转换的能力，在保证语义内容完整性的同时将文本尽可能转换成目标情感强度值。

细粒度文本情感转换具有一定的学术价值和应用价值。1）在学术研究方面，虽然近年来的文本情感转换研究工作具有较高的创新性和实用性，但对于细粒度文本情感转换任务存在研究关注度不足，效果不理想等问题。本课题发现通过依存句法分析可以捕捉文本中情感内容与其作用的主体对象之间的关系，辅助进行细粒度情感转换，在一定程度上是对文本风格转换领域的补充。同时，本课题首次将依存句法分析方法应用于细粒度文本情感转换任务上，具有一定的创新性。2）文本情感转换属于自然语言处理中自动文本生成技术的一种，自动化的文本生成可以大大增强人们的内容创作能力，替代部分重复性大、创造性低的工作。在实际应用方面，更细粒度的文本情感转换可以协助人们更好地进行内容创作，同时可以协助人们快速创作效果更为理想的文本。

## 1.3 文本情感转换国内外研究现状

文本风格转换是自然语言处理的热门研究和重难点。近年来，国内外的学者在机器学习顶级会议上，如ICML、AAAI、ACL等，发表了大量关于文本风格转换的研究工作。文本风格转换重点关注给定原句子和特定风格，对原始文本进行目标风格转换这一特定任务。大多数文本风格转换任务研究文本的情感这一风格属性[8][9][10][11]，也有少部分工作研究文本的书写风格[12][13]、正式非正式化[14][15]、时态[16]等风格属性。本课题主要研究文本情感，所以本节针对文本情感转换的现状进行分析。根据模型训练的方式，将现有方法分为文本情感转换的循环结构和非循环结构。由于本课题研究细粒度文本情感转换，虽然该课题现有研究较少，但也会在本节最后部分进行介绍。

### 1.3.1 文本情感转换的循环结构

为了使解码器生成指定风格属性的句子，引入鉴别器/分类器来指导解码器更好地工作。循环结构引入了对抗性（Goodfellow等人[17]）的思想，使解码器和鉴别器循环更新，其中鉴别器用于预测文本的风格属性。在每次迭代过程中，解码器使生成的文本不能被鉴别器正确预测，鉴别器尽最大可能去识别生成文本的属性。通过解码器和鉴别器之间的对抗，可以使得解码器和鉴别器都能达到比较好的效果。采用这种结构训练的具有代表性的工作如下。

在文本情感转换任务中，有很多研究是基于变分自动编码器（VAE）。变分自动编码器（Kingma和Welling等人[18]）是自动编码器升级版本中的一种，自动编码器是将输入编码成一个隐藏表示，然后再从隐藏表示中解码生成新文本。VAE与自动编码器的不同之处在于VAE的隐藏表示来自于训练期间学习到的概率分布，加入了推理的过程，因此编码器变成了一个变分推理网络。Yang等人[19]提出使用语言模型来取代传统二元分类器作为鉴别器，因为传统二元分类器所提供的误差信号是不稳定的并且不能训练解码器生成流畅的语言，而使用目标风格域的语言模型作为鉴别器可以提供更丰富更稳定的反馈，再结合VAE生成满足目标属性的文本。Hu等人[20]提出结合VAE和鉴别器来有效地生成语义表示，该模型将伪造的样本作为额外的训练数据采用唤醒-睡眠算法来提高VAE的性能。

后来研究(Bowman等人[21]，Yang和Salakhutdinov等人[22])发现，变分自动编码器在训练阶段容易发生训练崩溃的问题，并且VAE对文本建模和隐藏表示的后验分布没有抓住句子的内容信息，因此很多研究开始使用非变分自动编码器，比如使用自动编码器、翻译模型、对抗自动编码器等来取代变分自动编码器。Shen等人[23]将输入句子映射到一个与风格无关的内容表示，然后传递给与风格相关的解码器，使用两个鉴别器分别作用于两个属性域。Zhao等人[24]扩展了对抗自动编码器（AAE）（Makhzani等人[25]）以适应离散序列结构，该模型学习从输入空间到对抗性的连续潜在空间的编码器，不像AAE使用固定的先验，该模型学习参数的先验，并且不需要政策梯度或连续松弛。Fu等人[26]提出了两种模型用于文本风格转换，使用对抗性的方式进行训练，第一个模型使用多解码器模型（Sutskever，Vinyals和Le[27]），编码器用于捕获输入句子的内容表示，多解码器包含两个解码器来生成不同风格的文本。第二个模型使用相同的编码策略，但引入了风格嵌入，因此只需要一个解码器学习产生不同风格的输出。Shrimai等人[28]基于Rabinovich等人[29]的研究发现，可以利用反向翻译来改写句子并降低原有风格的影响，具体使用语言翻译模型来学习输入句子的潜在表示以保留句子的语义内容，减少风格属性的影响，然后使用对抗性的方式来生成符合所需风格的句子。Gong等人[30]提出使用强化学习的方式来训练模型，模型具有一个生成器和一个评价器，生成器是一个基于注意力机制的序列到序列模型，用以生成目标风格句子，评价器是一个鉴别器，由风格模块、语义模块、语言模型组成，用来给生成句子的风格、语义保存和流畅性打分。Huang等人[31]提出了循环一致对抗自动编码器（CAE），CAE包含三个基本组件：（1）LSTM自动编码器，可将文本编码为潜在的表示形式。（2）对抗性风格转换网络，使用经过对抗训练的生成器来转换潜在表示形式从一种风格到另一种风格的表示形式，以及（3）循环一致性约束，可增强转换网络内容保存的能力。

基于循环结构的情感转换方法存在模型训练崩溃、模型结构复杂等问题，并且大多数使用多个解码器作用于不同的情感属性，缺乏一定的灵活性。

### 1.3.2 文本情感转换的非循环结构

非循环结构没有引入对抗性的思想，分类器仅为模型提供反馈信号，指导解码器生成指定风格的句子，并用来预测生成句子的风格是否满足目标风格，作为指标衡量模型的效果。非循环结构有各种各样不同的形式，但总的来说还是使用编码器-解码器框架，再搭配其它的一些技术比如强化学习、记忆网络等，具有代表性的工作如下。

Li等人[32]提出通过去掉与句子风格相关的短语来提取内容信息，称为delete，再从目标风格语料库中检索相似的句子并提取其风格信息，称为retrieve，使用神经网络模型组合成一个最终输出，称为generate。他们在此基础上提出了四种模型，第一种模型仅仅包含delete和generate部分，第二种模型包含delete、retrieve、generate部分，第三种模型包含delete和retrieve部分，第四种模型仅包含retrieve部分。Igor等人[4]使用一个简单的编码器解码器框架和一个分类器，以及使用注意力机制和一套新的损失约束，分类器用于提供误差反馈指导解码器工作，注意力机制用于给编码器生成的隐藏表示添加权重，增加了反向转换损失用于约束模型生成内容完整和符合目标风格的句子。Zhang等人[33]提出使用记忆网络来辅助解码器，因为非风格上下文对风格词汇的出现提供了强有力的线索，记忆网络用于学习不同风格语料库的信息与上下文的记忆以此协助解码器生成指定风格的句子，并使用基于自注意力机制的分类器来将风格信息与内容信息分离。Zhang、Ding等人[34]提出共享-私有编码器解码器模型，共享模型学习所有实例的语义属性等公有属性，私有模型学习相应风格预料库的特定特征，通过结合共享模型与目标风格私有模型来生成目标句子。Xu等人[35]提出了循环强化学习模型，该模型包含两个部分：中和模块和情感模块，中和模块通过过滤掉情感内容来提取语义信息，情感模块将中和语义内容添加情感实现情感的转换，并使用强化学习的方式根据情感模块的反馈来奖励中和模块的输出。

后来，Transformer以及Bert两种更有效的语言模型被提出，Transformer和Bert在多项NLP任务中的表现超过了RNN、CNN，因此也逐渐地应用于文本风格转换。Transformer由Ashish Vaswani等人[36]在2017年发表的论文Attention Is All You Need中提出，只用编码器-解码器框架和注意力机制就能达到很好的效果，Transformer的编码部分由多个编码器构成，解码部分也由相同数量的解码器构成，可以高效地实现并行化。Dai等人[37]提出使用风格Transformer，不用对原始句子的潜在表示作出假设，使用Transformer中的注意力机制来更好地进行风格转换并保存其内容。Akhilesh等人[38]在Li等人[32]的基础上提出了使用delete Transformer和目标风格Transformer来提高其性能。Wu等人[39]发现情感转换过程与“文本填充”或“完形填空”的任务非常相似，可以由深度双向的“掩盖语言模型（MLM）”（例如BERT，由Devlin等人[40]2018年提出）来处理，因此提出了“掩盖和填充”两步法。在掩盖步骤中，通过掩盖情感标记的位置来将风格与内容分离。在填充步骤中，将MLM修改为“条件式MLM”，通过预测以上下文和目标情感为条件的单词或短语来填充掩盖的位置。

基于非循环结构的情感转换方法具有模型多样化、技术创新度高等特点，但很难适应于细粒度情感转换任务，需要对模型进行大幅度修改才能作用于细粒度情感转换。

### 1.3.3 细粒度文本情感转换

细粒度文本情感转换将情感标签从二元扩展为多元，该任务首先由Liao等人[41]提出。Liao提出了可量化序列编辑的任务：编辑输入序列以生成满足给定结果数值的输出序列，该结果数值可测量序列的某些属性，并要求保留输入序列的主要内容。具体地，在训练阶段，每个输入句子与数字结果相关联。例如，评论句子的结果是其评分，范围为1到5；每个广告的结果就是其点击率。在测试阶段，给定一个输入句子和一个指定的结果目标，模型需要编辑输入以生成一个新的句子，该句子需要满足结果目标。同时，输出语句应保留输入所描述的内容。为了实现该任务，Liao等人通过基于VAE的两个单独的编码器捕获潜在的内容因子和潜在的情感因子，并利用伪平行句子对的内容相似度和情感差异来增强模型区分两个因素的能力，然后在目标情感下修改内容。随着此任务的提出，Luo等人[42]也提出了结合情感强度值的Seq2SentiSeq模型，并使用循环强化学习方法进行训练，以此实现细粒度文本情感转换。

现有的细粒度情感转换方法相对较少，目前的研究对该领域的关注不足。并且现有的细粒度情感转换方法存在效果不好、生成文本质量差的问题。与他们不同的是，本文使用情感强度值作为额外控制的自动编码器，以及通过依存句法分析生成的伪平行句子对作为参考来实现对文本的修改，利用语言模型改善文本的流畅性。

## 1.4 论文的主要研究内容

针对细粒度文本风格转换任务，本论文主要进行以下内容的研究：

（1）依存句法分析辅助构造伪平行句子对的研究。由于观察到文本的情感内容与文本中的特定上下文具有紧密的关系，本课题研究使用依存句法分析对文本词汇间的依存关系进行分析，提取出情感信息与其修饰的上下文对象。通过使用提取出来的情感信息与特定上下文对象的关联关系来构造伪平行句子对，为模型提供训练信号，将无监督训练进一步转换为基于伪标签的半监督训练。

（2）基于依存句法分析的细粒度文本情感转换（FGSTDP）模型的研究。对于文本情感转换任务，现有的方法倾向于将文本的情感内容和语义内容分开，并在分开之后的语义内容上进行文本修改，这种方法会使得编码之后的中间表示损失部分语义信息；或者使用多个解码器作用于不同的目标情感，这对细粒度情感转换具有很强的局限性。由于现有方法的不足，本文针对细粒度文本情感转换任务，结合依存句法分析和多分类情感模型，提出了FGSTDP模型。该模型直接对原始文本进行修改，并使用约束条件使模型生成满足目标的文本。

（3）基于语言模型的细粒度文本情感转换（FGSTDP+LM）模型的研究。FGSTDP模型在文本情感转换中的效果优于现有方法，但存在生成文本流畅性较差的问题，因此FGSTDP+LM在FGSTDP的基础上，研究设计语言模型来改善生成文本的语言流畅性。

（4）设计实验证明模型的合理性和有效性。为了验证FGSTDP和FGSTDP+LM模型的有效性，将两个模型在公开数据集与对比方法进行对比，并使用BLEU、MED、MAE、Perplexity四个评价指标进行评估，并设计消融实验验证模型每一部分的合理性与有效性。

## 1.5 论文的组织结构

论文将根据下面六个部分分别对本文研究内容进行介绍：

第1章 绪论。介绍文本风格转换和情感转换的相关背景，引出本文的研究目的和意义，对国内外现状进行分析，简要概括本文的研究内容，交代论文的组织结构。

第2章 细粒度文本情感转换相关理论。介绍文本情感转换任务的相关理论基础和概念，包括依存句法分析、循环神经网络、编码器-解码器以及注意力机制。

第3章 基于依存句法分析的细粒度文本情感转换模型。介绍基于依存句法分析的细粒度情感转换模型的详细内容。即模型相关定义和总体结构，包括输入定义、输出定义等；描述情感内容提取相关内容和伪平行句子对构造方法；介绍基于编码器的多分类情感模型；设计基于依存句法分析的情感转换模型的结构；介绍基于依存句法分析的情感转换模型的训练过程；章节总结。

第4章 基于语言模型的细粒度文本情感转换模型。介绍基于语言模型的情感转换模型的详细内容，即模型相关定义和总体结构；描述模型训练前的准备工作；介绍基于双向GRU的语言模型；设计基于语言模型的情感转换模型的结构；介绍模型训练过程；对本章进行总结。

第5章 实验与分析。介绍实验环境及使用的数据集；描述实验评价指标及实验对比算法；介绍实验参数设置与方案设计；对实验结果进行展示与分析。

第6章 总结与展望。对本文的研究内容进行总结，包括创新点与结果分析，针对本文研究工作的不足与改进提出后续的研究方向。

# 2 相关理论基础

本文为了解决文本细粒度情感转换问题，提出了FGSTDP和FGSTDP+LM模型。为了方便后续相关内容的阅读与理解，本章将介绍与FGSTDP和FGSTDP+LM模型相关的理论基础。首先介绍依存句法分析的相关内容，然后介绍循环神经网络相关模型；最后介绍编码器-解码器架构，以及注意力机制。

## 2.1 依存句法分析

2.2.2这是一个伪装的三级标题

2 是否识别为标题

2 标题？

2，标题？

2．标题？

3.3是否识别为标题。句法分析是根据某种语法系统（中文/英文语法系统等）将句子中的句法关系（主谓宾、定状补等）表述为图结构，依存指的是句子中单词之间具有方向性的支配与被支配关系。支配的单词视为支配对象（Head），被支配的单词视为从属对象（Dependency）。依存句法分析就是根据句子的句法结构，分析各个成分之间的依赖关系，依赖关系包含主谓关系、动宾关系、形容词修饰语等。依存句法分析将句子中具有句法关系的两个词使用依赖弧连接起来，每个依赖弧表示一个依赖关系，箭头指向支配对象，箭头的起点是从属对象，如图2.1所示。依存语法分析因为其简单的表示形式以及易于标注易于使用而吸引了越来越多的目光。依存句法分析是自然语言理解的重点研究任务之一，驱动了多种研究任务的发展，比如信息提取、情感分析和机器翻译。



图2.1 依存句法分析示例

常用的依存句法分析方法有基于图的依存句法分析方法、转移的依存句法分析方法以及联合模型的依存句法分析方法。基于图的依存句法分析从左到右分析句子构成完全有向图，然后从中找到最大的生成树。图中每条边连接着两个单词，边的权重表示这两个单词存在某种依存关系的可能性。基于图的依存句法分析方法中最经典的模型是Timothy Dozat等人[43]提出的双仿射注意力机制模型（Biaffine Attention），如图2.2所示。该模型建立了一个规模更大但使用更多规则化的网络，将传统的基于多层感知机（MLP）的注意力机制和仿射标签分类器替换为biaffine分类器，在基于图的简单依存解析器中使用神经注意力，以此预测单词间存在某种依存关系的概率。基于转移的依存句法分析方法将构造语法树的过程定义为一个有限自动机问题，每一步根据当前状态预测转移到下一个状态的动作，通过一系列转移动作（移进、左规约、右规约和根出栈）构造依赖项语法树，学习的目的是找到最佳动作序列。基于转移的依存句法分析方法最好的模型是Dyer C等人[44]提出的Three Stack LSTM模型，如图2.3所示。该模型使用三种堆栈LSTM：一种代表输入，一种代表部分句法树的堆栈，一种代表对解析状态进行编码的解析动作的历史记录，由于部分句法树的堆栈可能同时包含单个标记和部分句法结构，因此使用递归神经网络来组成单个树片段的表示。基于联合模型的依存句法分析方法将词性标注或者分词与句法分析结合，利用词性信息和词信息辅助进行句法分析。现有的很多工具都提供依存句法分析的功能，比如由斯坦福大学开发的StanfordCoreNLP[[1]](#footnote-1)，Python高级自然语言处理库Spacy[[2]](#footnote-2)，复旦大学自然语言处理实验室开发的中文NLP工具包FudanNLP[[3]](#footnote-3)等。

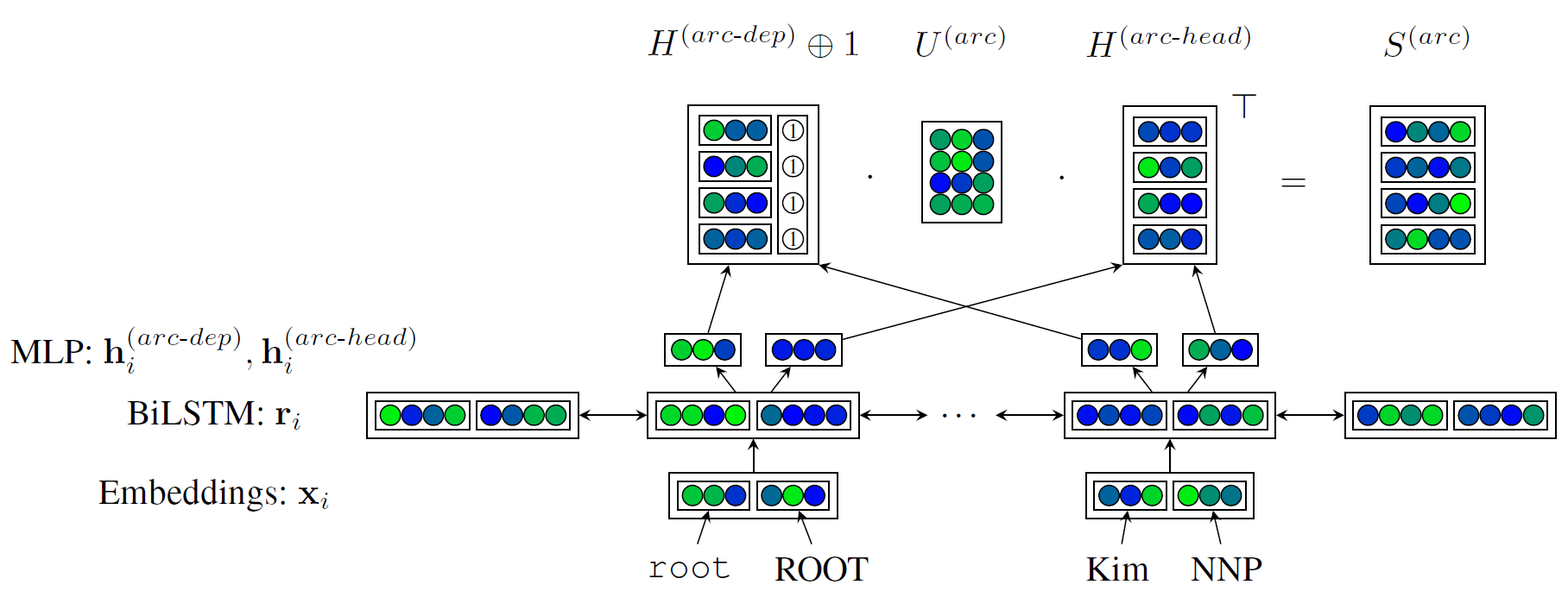


图2.2 Biaffine Attention模型结构图



图2.3 Three Stack LSTM模型结构

## 2.2 循环神经网络

深度学习的主流模型之一的循环神经网络（RNN[45]），该网络随着时间的推移对序列数据进行循环计算，即当前时刻的隐藏变量由当前输入和上一时刻的隐藏变量共同决定。循环神经网络就像人脑一样拥有记忆，能够捕获序列的历史信息，而不像其它神经网络不能对序列的历史信息留存记忆，因此循环神经网络广泛应用于文本序列、图形图像等需要进行时间序列分析的应用中。



图2.4 循环神经网络的结构图

循环神经网络的整体架构如图2.4所示，图中左边是该网络的简单构造，包括输入层、输出层、隐藏层。其中是输入的向量表示，是隐藏状态的值，是输入层到隐藏层的权重，是输出的向量表示，是隐藏层到输出层的权重，是上一隐藏状态的值到当前隐藏状态的权重。现在将模型按时间线展开，如图2.4右边所示，图中表示时刻t的隐藏状态的值，表示时刻的隐藏状态的值，表示时刻的隐藏状态的值。从图中可以看到的值不仅仅取决于当前输入，还取决于，的计算可以由公式2.1得到：

|  |  |  |
| --- | --- | --- |
|  |  | (2.1) |

为非线性激活函数，比如双曲正切函数（Tanh）或者线性整流函数（ReLU）。当前的输出的计算由公式2.2得到：

|  |  |  |
| --- | --- | --- |
|  |  | (2.2) |

为激活函数，将公式2.1代入到公式2.2后可以得到公式2.3：

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  | (2.3) |

RNN使用基于时间顺序的反向传播方法求解模型参数。对于RNN来说只能捕获短距离依赖关系，因为当序列较长时，序列后面的梯度很难反向传播到前面，会发生梯度消失的问题。

为了解决传统循环神经网络不能处理长距离依赖缺陷，长短期记忆网络（Long Short Term Memory，LSTM）被Hochreiter等人[46]提出。该网络引入门机制，包括输入门、输出门和遗忘门，以及与隐藏状态具有相同形状的记忆细胞，LSTM的网络结构图如图2.5所示。原始RNN的结构是一种较为简单的重复链式结构，LSTM同样具有这种链式结构，但包含四个重复的模块，结构稍微复杂一些。LSTM的关键部分是细胞状态，即与隐藏状态相似的记忆细胞，在整个链式结构上贯穿运行。细胞状态之间是简单的线性组合，信息在它们之间保持不变。LSTM设计门控结构来去除或者增加信息到细胞状态，门控结构主要包含一个Sigmoid神经网络层和一个按位乘法（pointwise）操作。Sigmoid层将输出变换到0到1之间，“0”表示丢弃全部信息，“1”表示信息全部保留。LSTM中的遗忘门决定历史细胞状态的舍弃与保留，LSTM的输入门决定将新信息选择性地保留到细胞状态中，其中Sigmoid层决定将要更新的值，Tanh层创建一个新的候选记忆细胞。LSTM的输出门决定最终的输出值。LSTM设计三个门控单元，梯度能够顺利地在网络中传递更新参数，很大程度上降低了梯度消失的概率。



图2.5 LSTM网络结构图

Cho等人[47]提出了一种LSTM的有效变体：门控循环单元（Gated Recurrent Unit，GRU），GRU合并遗忘门和输入门为更新门，同时将隐藏状态和细胞状态混合。所以GRU只设计了更新门和重置门两个门控，其网络结构图如图2.6所示。GRU的重置门决定了现在时刻的候选隐藏状态如何被上一时刻的隐藏状态所更新，用于决定保留还是丢弃历史信息。更新门决定了当前时刻的候选隐藏状态如何去更新隐藏状态。GRU的这种设计可以解决梯度消失的问题，并且能捕捉到时间步距离较大的依赖关系。GRU相对于LSTM结构更简单，因此GRU参数更少，训练速度更快，并且其效果可以和LSTM比拟，所以本文使用的循环神经网络主要是GRU模型。



图2.6 GRU结构图

## 2.3 编码器-解码器结构与注意力机制

编码器-解码器（encoder-decoder）结构主要用于处理输入输出为不定长序列的情况，sequence-to-sequence（seq2seq）模型属于编码器-解码器结构的一种，输入输出都是序列，最早由Sutskever等人[48]提出，主要用于机器翻译。以“英译汉”为例，输入是一句英文序列，输出为对应的不定长的中文翻译序列，比如“machine learning”翻译为“机器学习”，输入长度为2，输出长度为4。Seq2seq模型不仅用于机器翻译领域，还广泛地应用于问答系统、文本摘要生成、对话生成等领域。编码器-解码器的基本结构如图2.7所示，编码器和解码器是两个神经网络，可以是循环神经网络（RNN、LSTM、GRU）、多层感知机（MLP）等。编码器用来分析输入序列，将输入序列表示为特定长度的语义编码，解码器用于生成序列，将中间语义编码解码得到输出序列。



图2.7 编码器-解码器结构

为了方便描述编码器-解码器结构，假设编码器和解码器都是用的循环神经网络。在编码器中，输入序列被表示成词向量（word embedding），假设输入序列为，是输入序列中的第个词，在编码器中被表示为词向量。在时间步为时，当前时刻的隐藏状态由当前输入的向量表示和上一时刻的隐藏状态变换得到，见公式2.4，为编码器隐藏层的变换函数。然后编码器通过自定义函数将隐藏状态转换为中间语义表示，见公式2.5。假设输出序列为，当时间步为（与编码器的时间步不同）时，当前输出的条件概率由所有的输出序列和中间语义表示决定，即。解码器当前时刻的隐藏状态由上一时刻的隐藏状态、中间语义表示、上一时间步的输出转换而来，见公式2.6，其中是解码器隐藏层变换函数。有了当前时间步的隐藏状态之后就可以使用输出层及softmax运算来得到当前输出的条件概率。

|  |  |  |
| --- | --- | --- |
|  |  | (2.4) |
|  |  | (2.5) |
|  |  | (2.6) |

在基础编码器-解码器结构中，解码器每一时间步的输入都是上一时间步的输出，所以当某一时间步的解码结果不好时，会影响后续的解码效果，就会使得训练结果误差越来越大。因此本文中的解码器训练时不使用传统的free-running模式，而是使用teaching forcing模式，teaching forcing模式的流程图如图2.8所示，使用训练数据的标准答案对应的上一项作为当前解码器的输入。预测第一个词汇时输入为“\_GO”，直到输出为“\_EOS”时解码结束。



图2.8 teaching forcing流程图

由于模型测试时测试数据没有标准答案，因此测试时无法对模型使用teaching forcing模式。为了保证测试时生成序列的效果和模型的跨域能力及性能，本文在模型测试阶段使用集束搜索（beam search）来生成句子。在基础编码器-解码器结构中，每个时间步解码器的输出都是选择使得条件概率最大的词。而集束搜索在每个时间步选择使得条件概率最大的个词，与之前的输出组成个候选输出序列，在之后的每个时间步都基于之前的个候选输出序列组合成个输出序列，为词典大小，然后选出个最佳的序列作为该时间步的候选输出序列。最终从最后时间步的候选输出序列中选出最优的作为模型的输出。

在经典编码器-解码器结构中，解码器在每个时刻通过相同的中间语义表示来获取输入序列的信息，所以解码器在生成每一个输出词汇的时候都是用相同的中间语义表示，输入序列中的每一个元素对解码器每个时间步的输出的贡献是相同的。然而输入中的每一个词汇对当前输出的贡献都是不一样的，比如输入为英语序列“machine learning”，输出为中文序列“机器学习”，当翻译“机器”这个单词的时候，原始输入中的每个单词对翻译目标单词“机器”的贡献明显不同，“machine”这个词汇的贡献更大。因此解码器在生成输出序列的每一个词的时候只需要利用原始输入的部分信息，在每一时间步使用的中间语义表示也应该是不同的。为了应对这个问题，本文引入了注意力机制，注意力模型借鉴了人类的注意力机制。在解码生成每一个输出单词的时候对输入序列的编码信息分配不同的注意力，该想法通过编码器在每一时间步调整中间语义表示各个部分的注意力权重来实现。加入注意力机制的编码器-解码器结构如图2.9所示，图中的背景变量就是上文提到的中间语义表示。



图2.9 编码器-解码器上的注意力机制

每一时间步的中间语义表示的计算如公式2.7所示，其中是编码器的时间步，是解码器的时间步，为编码器在时间步的隐藏状态，是注意力权重，计算方式如公式2.8所示。的计算取决于解码器上一时间步的隐藏状态以及编码器在当前时间步的隐藏状态，见公式2.9，函数可以是不同的选择，比如内积运算。

|  |  |  |
| --- | --- | --- |
|  |  | (2.7) |
|  |  | (2.8) |
|  |  | (2.9) |

如果编码器-解码器使用的是GRU模型，则可以对其设计稍作修改，加入含有注意力机制的中间语义表示。重置门计算如公式2.10所示：

|  |  |  |
| --- | --- | --- |
|  |  | (2.10) |

更新门计算如公式2.11所示：

|  |  |  |
| --- | --- | --- |
|  |  | (2.11) |

候选隐藏状态的计算如公式2.12所示：

|  |  |  |
| --- | --- | --- |
|  |  | (2.12) |

其中和分别是门控单元的权重参数和偏置参数。

## 2.4 本章小结

本章介绍了细粒度文本情感转换的基础理论知识。第一部分介绍了依存句法分析的原理，以及常用的模型和工具。第二部分介绍了循环神经网络RNN、LSTM、GRU模型，详细探讨了它们的网络架构。第三部分介绍了编码器-解码器的结构，由于原始free-running训练模式的缺陷，引入了teaching forcing模式来训练模型，并使用beam search在测试阶段指导解码器生成文本。然后引入注意力机制使得解码器在不同阶段关注有用的输入信息，保证了模型的性能与跨域能力。

# 3 基于依存句法分析的细粒度文本情感转换模型

上一章介绍了细粒度文本情感转换任务的相关理论与基础，针对已有研究的局限与不足，本文提出了基于依存句法分析的细粒度情感转换模型（FGSTDP）。本章将对该模型进行详细介绍，首先描述细粒度情感转换模型的基本定义，然后介绍模型训练前的准备工作，即情感内容提取与伪平行句子对构造。再分别详细阐述模型的具体结构及训练过程，最后对本章内容进行总结。

## 3.1 模型基本定义和总体结构

文本细粒度情感转换主要是在保留原始语义的基础上，对原始文本进行改写以满足目标情感值。整个模型的输入用来表示，，其中表示一个文本序列，是该文本的情感标签（，是细粒度情感值，范围为1到5）。本文规定情感值大于3的文本为积极文本，情感值小于3的文本为消极文本，情感值为3的文本为中性文本。细粒度情感转换任务的目标是生成一个新的文本，该文本对应的原始文本为，的情感标签为（）。生成的文本需要满足给定的目标情感值（），并与表达的语义内容相同。为了实现该目标，本文提出了FGSTDP模型，图3.1为FGSTDP模型的总体结构。图中上方是依存句法分析模块，该模块包含依存句法分析和伪平行句子对构造两部分。依存句法分析用来查找与句子中的情感词具有特定依存关系的上下文，该方法包含两个步骤：（1）提取输入句子的情感内容（2）分析句子中单词之间的依存关系，找到情感词的特定上下文词汇。下方是情感转换模块，也是整体结构的主要部分，该模型采用经典的编码器-解码器结构，输入为带有情感标签的文本，以及目标情感值作为解码器的额外输入进行情感控制。情感转换模型的目标是生成新的输出句子使得约束最小，FGSTDP模型使用了四个约束条件。

## 3.2 情感内容提取

在进行依存句法分析之前，需要找到文本中具有强烈情感极性的内容。根据观察



图3.1 FGSTDP模型总体结构图

可以发现，积极文本中的情感内容出现在消极语料库中的概率很小，而与情感无关的语义内容则会很大概率也出现在消极语料库中，反之亦然。比如在“the food is delicious”中，“delicious”是该文本相应的情感内容，该词汇在积极语料库中经常出现，在消极语料库中出现的次数很少。而该文本中剩余的与情感无关的内容“the”、“food”、“is”不管是在积极语料库还是在消极语料库中出现的频率都很高。因此根据观察到的这个现象，可以使用一种简单的方法来提取文本中的情感内容，该方法参考Li等人[32]的做法。在此处，情感值不需要进行细粒度划分，仅仅只考虑文本的二元情感属性，即消极和积极属性。本文假设所有输入文本集合为，表示文本表达的情感极性（）。输入句子由N个词构成，，的情感极性为。计算中每个词汇在相应情感极性语料库和相反情感极性语料库中出现的相对频率，计算方式如公式3.1所示：

|  |  |  |
| --- | --- | --- |
|  |  | (3.1) |

其中表示单词在相应情感极性语料库中出现的次数，表示单词在相反情感极性语料库中出现的次数，为平滑参数。如果单词的相对频率大于设定阈值，则单词被认定为输入句子的情感内容，设定为文本的全部情感内容。

## 3.3 伪平行句子对构造

### 3.3.1 依存句法分析

在提取文本中的情感内容之后，对文本进行依存句法分析以找到与情感内容有特定依存关系的上下文词汇。依存句法通过每个词汇间的依存关系来表达整个句子结构，这些依存关系构成了一棵依存语法树，其根节点是句子的核心谓词。根据语法树中的依存关系，我们可以在句子中找到两个具有特定依存关系的单词，它们通常不相邻。如图3.2所示，图中每个箭头表示一个依存关系，箭头指向支配对象，箭头的起点是从属对象。要确定文本中哪个上下文词与情感词具有特定的依存关系，本文只考虑几个固定的依存关系，例如名词主语（nsubj），直接对象（dobj），形容词修饰语（amod）等。例如，句子“the food tastes delicious”中的“food”是我们想要找到的与情感词“delicious”有特定依存关系的上下文词，而不是“tastes”或其它词，“food”一词是情感词“delicious”的名词主语nsubj。



图3.2 依存句法分析

假设是原始文本的情感词汇，，本文设定是文本中与有特定依存关系的上下文词汇。由于此处不需要考虑情感的细粒度划分，我们提取不同情感极性（positive、negative）语料库中所有文本的情感词与特定上下文()。例如积极情感文本“the food tastes delicious”和“the food tastes wonderful”都是描述“food”，因此情感词汇“delicious”和“wonderful”都被存储起来作为“food”的相关修饰形容词。情感词和相应的上下文词之间的依存关系将用于辅助构造伪平行句子对，在下面一小节中进行详细描述。

### 3.3.2 情感内容替换

伪平行句子对是具有相同语义内容但表达的情感粒度不同的句子，如表3.1所示。构造伪平行句子的方法是用另一个情感词替换原始文本的每个情感词。如上所述，情感词与其特定上下文词具有密切的相关性，因此其上下文词的所有相关情感词都可以作为替换原始情感词的候选词。

表3.1 伪平行句子对

|  |  |
| --- | --- |
| 情感值 | 对应文本 |
| 1 | the worst part, dreadful service and prices can not be beat! |
| 2 | the frustrating part, lousy service and prices can not be beat! |
| 3 | the hot part, fine service and prices can not be beat! |
| 4 | the best part, exceptional service and prices can not be beat! |
| 5 | the gorgeous part, wonderful service and prices can not be beat! |

给定输入，是的一个情感词汇，，是与具有特定依存关系的上下文单词。假设形容的有k个候选词汇可以用来替换，在目标情感值下选择最佳的候选词来替换，选择的词汇需要满足使得公式3.2的值最小。

|  |  |  |
| --- | --- | --- |
|  |  | (3.2) |

是加权记分器，是情感词汇的所有候选替换词汇。记分器从各个不同的角度来衡量每个候选词汇，在本文中主要考虑两个因素：（1）候选词表达的情感力度与目标情感值的差异（2）候选词与原始情感词的相似性。本文使用下述方式对候选词进行评估：

**情感值差异**：情感值差异是指候选词表达的情感值与目标值之间的差异。 由于的情感值未知，因此如何计算的情感值是一个关键问题。受sendiWordNet [1]的启发，本文使用包含单词c的所有文本的情感值的平均值来表示c的情感，如公式3.3所示：

|  |  |  |
| --- | --- | --- |
|  |  | (3.3) |

其中是输入句子，，是的情感值，是情感词c出现在语料库中的文本总数。然后情感词的情感粒度与目标情感值的差异计算方式如公式3.4所示：

|  |  |  |
| --- | --- | --- |
|  |  | (3.4) |

**相似性**：相似性衡量了情感词（）和候选词（）之间语义的相似度。如观察所示，所有候选词汇都可以替换，但是某些候选词汇与原始情感词上下文不匹配。例如，文本“the food is delicious”中的情感词“delicious”更可能由“wonderful”代替而不是“helpful”。因此，需要找到一个与原始情感词表达相似语义的来替换，词汇间的语义相似性计算如公式3.5所示：

|  |  |  |
| --- | --- | --- |
|  |  | (3.5) |

其中使用的是单词和嵌入向量之间的余弦相似度来计算，本文使用word2vec模型（由Tomas等人[49]提出）来计算词汇间的语义相似性。word2vec是用来构建词向量的模型，将每个词表示为一个向量，可用来表示词与词之间的关系。word2vec训练的词向量是低维、稠密的，word2vec利用了词的上下文信息获得丰富的语义信息。word2vec有两个重要模型，CBOW模型和skip-gram模型。CBOW模型是通过某个词相邻的n个词来计算该词出现的概率，skip-gram模型是根据某个词分别计算它相邻的每个词汇出现的概率。本文使用word2vec模型的skip-gram结构，用所有输入文本对模型进行训练，使用训练好的模型来生成词向量，然后计算词向量的余弦相似度。

计分器函数由上述所有度量组成，如公式3.6所示：

|  |  |  |
| --- | --- | --- |
|  |  | (3.6) |

其中和是权重参数，伪平行句子构造算法如算法3.1所示。

|  |
| --- |
| **算法3.1** 基于依存句法分析的伪平行句子构造算法 |
| **输入:** 输入句子，情感值标签为，目标情感值，上下文词-情感词的词表  **输出:** 句子的基于情感值的伪平行句子  1: 基于公式3.1提取句子的情感词汇  2: 对句子中词汇间的依存关系进行分析  3: **for** in A **do**  4: 根据R找到与有特定依存关系的非情感上下文词汇  5: 在表中查找与上下文词汇有特定依存关系的所有情感词作为的候选替换词  6: 用在表中更新与有关的情感词  7: 基于公式3.3计算中所有候选情感词的情感值  8: 基于公式3.4计算中每个候选情感词与目标情感值的情感值差异  9: 基于公式3.5计算中每个候选情感词与的相似性  10: 基于公式3.6使用记分器函数找到中最佳的替换词汇  11: 使用候选词替换原始情感词，得到目标情感值下的原始文本的伪平行句子  12: **end for** |

## 3.4 基于编码器的多分类情感模型

为了让模型能将文本的原始情感转换为目标情感，文本情感转换会引入分类器来指导模型按正确的情感方向转换。粗粒度文本情感转换只涉及消极和积极两种情感，使用二分类情感模型，而细粒度文本情感转换涉及多个情感值，因此需要使用多分类情感模型。为了能更准确地提取序列特征，本文首先使用单向GRU编码器对文本序列进行特征提取，然后再将特征向量送入多分类模型，多分类模型由简单的两层全连接层组成，最后一层为输出层，一共有5个输出单元，对应5个类别，结构如图3.3所示。



图3.3 多分类情感模型

输入文本首先经过编码器的embedding层，得到每个词汇的词嵌入向量，如公式3.7所示。然后将词嵌入向量作为单向GRU的输入，得到特征向量，如公式3.8所示，其中和是权重参数，是偏置参数。

|  |  |  |
| --- | --- | --- |
|  |  | (3.7) |
|  |  | (3.8) |

分类器使用简单的两层网络结构，第一层为全连接层，使用ReLU激活函数，是权重参数，是偏置参数。最后一层为只有5个神经元的全连接层，输出文本的概率分布，如公式3.9-3.10所示。

|  |  |  |
| --- | --- | --- |
|  |  | (3.9) |
|  |  | (3.10) |

多分类情感模型使用所有输入文本及其情感标签来训练，多分类模型的预测值与真实情感值的差异，如公式3.11所示，作为指导信号来训练多分类模型。

|  |  |  |
| --- | --- | --- |
|  |  | (3.11) |

## 3.5 基于依存句法分析的情感转换模型结构

FGSTDP使用的情感转换模型主要采用编码器-解码器框架，模型结构如图3.4所示。文本序列首先经过Embedding层，得到文本的词嵌入向量，如公式3.12所示。然后将文本词嵌入向量作为编码器的输入，编码得到文本的中间语义表示，如公式3.13所示。编码器主要用于提取文本的语义、句法等特征，采用单层双向GRU模型。编码得到的中间语义表示将作为解码器的输入，除此之外，解码器还有一个额外输入，情感控制嵌入，用以控制情感转换方向。编码器解码器之间加入注意力机制，给编码器的输出加上注意力权重。解码器也使用单层双向GRU模型，解码器训练模式采用teaching forcing模型，即在每一时间步使用当前时间步对应的真实数据作为输入，而不是上一时间步的输出。因此解码器在每一时间步的输入包括编码器输出、情感控制嵌入、当前时间步的输入，这三个输入首先经过一层全连接层进行合并，然后再经过GRU层得到当前时间步的输出，如公式3.14所示。最后解码器输出经过一个全连接层得到最终输出，即词汇表中单词的概率分布，如公式3.15所示。

|  |  |  |
| --- | --- | --- |
|  |  | (3.12) |
|  |  | (3.13) |
|  |  | (3.14) |
|  |  | (3.15) |

其中，和分别是编码器、解码器的隐藏层权重参数。和分别是编码器、解码器输入层和隐藏层的权重参数，是解码器输出层权重参数，、和是偏置参数。表示dropout层，以一定概率丢弃输入的部分内容，用于防止过拟合。是编码器解码器注意力层，用于给编码器输入加入注意力权重。



图3.4 基于依存句法分析的情感转换模型结构

## 3.6 基于依存句法分析的情感转换模型训练

在FGSTDP模型训练时，将带有情感标签的自然语言文本作为模型的输入，并使用约束条件控制训练的梯度方向，如图3.5所示。编码器学习将句子编码为隐藏的表示形式，而解码器学习在该表示形式下生成句子。但是，解码器生成的句子是类似于输入的新文本，解码器无法为其改写情感。因此，本文在模型中引入情感控制和一些约束条件，情感控制是目标情感值的嵌入向量，它与隐藏表示形式一起作为解码器的输入。约束条件是一组损失函数，这些损失函数用于提高模型的内容保存和情感转换的能力。本文引入分类器来预测生成句子的情感值，分类器在模型训练之前训练，训练好之后保持不变。定义编码器-解码器结构为，定义分类器模型为C。本文考虑下面四个损失，重建损失和反向重建损失被用来控制句子语义内容的完整性。此外，除了有助于句子语义内容的完整性，参照损失还有助于修改情感内容。

（1）**重建损失**：重建损失表示重新构建输入句子的误差，假设是输入句子经过编码器编码的隐藏表示，是的情感值。使用解码器在输入为和时生成句子，重建损失的计算如公式3.16所示：



图3.5 FGSTDP模型训练结构

|  |  |  |
| --- | --- | --- |
|  |  | (3.16) |

（2）**反向重建损失**：假设在目标情感值下转换之后的文本为，现在使用生成文本作为转换模型的输入，用来重建输入文本。是文本经过编码器编码之后的隐藏表示，将和作为解码器的输入，生成的文本为。反向重建损失为使用反向重建的误差，计算方式如公式3.17所示：

|  |  |  |
| --- | --- | --- |
|  |  | (3.17) |

（3）**分类器损失**：分类器用来预测文本的情感值，为了保证生成文本的情感值满足目标情感值，本文使用分类器损失作为反馈信号来引导模型。分类器预测的生成文本的情感值为，分类器损失的计算如公式3.18所示：

|  |  |  |
| --- | --- | --- |
|  |  | (3.18) |

（4）**参照损失**：参照损失是生成文本与在目标情感值下的伪平行句子之间的差异，计算方式如公式3.19所示：

|  |  |  |
| --- | --- | --- |
|  |  | (3.19) |

在训练中，使用句子和相应的情感标签作为输入以及预测情感值作为输出来训练分类器。在多次迭代之后，对分类器进行训练以最大程度地减少损失，然后将其添加到编码器-解码器框架中，并在后续模型训练中保持不变。通过最小化公式3.20来训练编码器-解码器网络，将原始文本、目标情感作为模型输入，伪平行句子作为参考以及新句子作为输出：

|  |  |  |
| --- | --- | --- |
|  |  | (3.20) |

## 3.7本章小结

本章针对现有方法的不足与局限，并结合观察发现，提出了基于依存句法分析的细粒度情感转换模型FGSTDP。本章对FGSTDP模型进行了详细介绍，首先描述模型的基本定义和总体结构。接下来对模型训练前的准备工作进行阐述，介绍了情感内容提取和伪平行句子对构造过程。然后介绍基于编码器的多分类情感模型，该模型用以预测生成句子的情感值，并参与模型训练过程，为模型提供反馈。最后介绍了FGSTDP模型的结构以及训练过程。

# 4 基于语言模型的细粒度文本情感转换模型

FGSTDP模型生成的句子存在情感词重复、语义不通的问题，为了提升生成句子的流畅性，本文在FGSTDP模型的基础上使用预训练语言模型为生成器提供更直接的训练信号，来改善生成句子的可读性，提出基于语言模型的细粒度文本情感转换模型FGSTDP+LM。

## 4.1 模型基本定义和总体结构

本章提出的FGSTDP+LM模型的相关定义与FGSTDP模型相似。模型的输入用来表示，，表示输入文本序列，为对应情感标签。，表示细粒度情感值，范围为1到5。同样规定情感值大于3的文本为积极文本，情感值小于3的文本为消极文本，情感值为3的文本为中性文本。本章模型的目标也是针对细粒度情感转换任务，即修改原始文本（情感标签为，）为一个新的文本，使得生成的文本满足给定的目标情感值（），并与具有语义内容相同。FGSTDP+LM模型的总体架构和FGSTDP模型相似，如图4.1所示。FGSTDP+LM模型总体也分为两个部分：伪平行句子对构造和情感转换。伪平行句子对构造部分和FGSTDP模型的相同，包括依存句法分析和伪平行句子对构造两部分。情感转换部分较FGSTDP模型多了语言模型损失约束，因此FGSTDP+LM模型的目标是生成新的句子使得图中的五个损失函数最小。

## 4.2 训练前的准备工作

在FGSTDP+LM模型训练之前，也需要和FGSTDP模型一样进行情感内容提取和伪平行句子对构造。情感内容提取需要识别文本中与情感相关的词汇，这里采用的方法和3.2节相同。通过计算句子中每个词汇在相应情感极性语料库和相反情感极性语料库中出现的相对频率来提取情感内容，当相对频率大于某个阈值时，该词汇被认定为情感内容。当句子的情感内容被提取之后，需要找到句子中与其具有特定依存关系的上下文，这里也采用与3.3节相同的方法，通过依存句法分析找到特定的上下文，



图4.1 FGSTDP+LM模型总体结构

然后通过情感词替换的方式来构造伪平行句子对。构造的伪平行句子在模型训练期间为模型提供反馈信号，同时引入多分类情感模型预测生成句子的情感值，并计算与目标情感值的差异来指导模型训练，此处的多分类情感模型使用3.4节的基于编码器的多分类情感模型。为了改善生成句子的语言流畅性，FGSTDP+LM模型提出使用语言模型来控制模型生成句子的可读性，语言模型在模型训练期间通过预测生成句子的流畅性，为模型提供反馈信号，语言模型将在下一节详细阐述。

## 4.3 基于双向GRU的语言模型

语言模型通过对语言序列的概率分布进行建模，以此评判一个新句子是否为正常语句，即新句子的概率分布是否与语料库中句子的概率分布相同。语言模型分为统计语言模型（如HAL模型[50]、LSA模型[51]、COALS模型[52]）和神经网络语言模型（如FFNNLM模型[53]、RNNLM模型[54]），由于统计语言模型具有维度稀疏和泛化性能差的缺点，Bengio等人[53]提出使用前馈神经网络来替代统计方法，神经网络语言模型解决了统计语言模型维度稀疏的问题并且具有很好的泛化性能。神经网络语言模型分为前馈神经网络语言模型和循环神经网络语言模型，前馈神经网络语言模型利用词向量映射解决了维度灾难，但对上下文信息的处理具有局限性。而循环神经网络语言模型通过门控单元来处理序列的长期依赖关系，可以很好地捕捉上下文信息。因此本文使用双向GRU语言模型对训练数据中的句子序列进行建模，学习句子序列的概率分布。在情感转换模型训练阶段，对生成句子的概率进行预测，将预测信息传递到生成器，控制生成器生成句子的流畅性。双向GRU语言模型的结构如图4.2所示。



图4.2 双向GRU语言模型结构

假设有一个输入序列，语言模型的目的是根据输入去推测，即根据句子前n-1个词汇去推测最后一个词汇。首先序列经过语言模型的embedding层，得到每个词汇的词嵌入，计算如公式4.1所示。再将词嵌入作为双向GRU模型的输入，得到中间隐藏表示作为输出层的输入，计算如公式4.2所示。最终的输出计算如公式4.3所示，表示是字典中每个单词的概率。

|  |  |  |
| --- | --- | --- |
|  |  | (4.1) |
|  |  | (4.2) |
|  |  | (4.3) |

其中、、表示权重参数，和为偏置参数。为了能为生成器提供可靠的训练信号，首先需要重新处理训练数据以适应语言模型，处理方式为将每个文本序列分成两部分，一部分为去掉最后一个单词之后的序列，一部分为去掉第一个单词的序列，前者作为语言模型的输入去预测后者。一般使用困惑度[55]（Perplexity）来评估语言模型，测试数据都是符合相似概率分布的句子，困惑度越小，句子的概率越大，语言模型性能越好，困惑度计算如公式4.4所示。预训练的语言模型在后期情感转换模型训练期间保持不变，使用困惑度去评估生成句子的流畅性。

|  |  |  |
| --- | --- | --- |
|  |  | (4.4) |

## 4.4 基于语言模型的情感转换模型结构

基于语言模型的细粒度文本情感转换模型在FGSTDP模型的基础上加入了语言模型，模型结构如图4.3所示。模型总体结构由一个编码器、一个生成器（编码器）、一个多分类情感模型、一个预训练的语言模型组成。自然语言序列以及其情感值标签作为模型的输入，得到文本的中间语义表示，计算方式如公式4.5所示。编码器主要用于提取文本的特征，如语义特征、结构特征等。得到的特征向量将会传入生成器，生成器还有一个额外的输入，即目标情感值，用以控制情感修改方向，修改之后的文本为，计算如公式4.6所示。

|  |  |  |
| --- | --- | --- |
|  |  | (4.5) |
|  |  | (4.6) |

为了使生成的文本满足要求，本文引入了重建和反向重建过程。图4.3的上半部分为重建过程，旨在优化模型还原原始文本的能力，以及优化编码器提取文本特征的能力。在此过程中，生成器的额外输入为原始文本的情感值，生成的文本与原始文本的差异定义为重建损失。图4.3的下半部分为反向重建过程，旨在优化模型保留原始语义内容的能力，以及优化生成器生成文本的性能。在此过程中，生成文本将作为编码器的输入，作为生成器额外情感值输入，用以还原原始文本，生成的文本与的差异定义为反向重建损失。重建损失和反向重建损失主要用来让生成文本与原始文本具有相同的语义内容，以此保留文本语义信息。为了让生成文本具有目标情感值，将作为分类器的输入以预测的情感值，预测值与目标值的差异定义为分类损失。本文还将生成文本与目标情感值对应的伪平行句子之间的差异定义为参照损失，以此加强模型保留语义内容和转换准确率的能力。语言模型用来控制生成文本的流畅性，语言模型对生成文本预测的概率分布与原始文本分布之间的差异定义为语言模型损失。



图4.3 基于语言模型的情感转换模型结构

## 4.5 基于语言模型的情感转换模型训练

首先训练语言模型和多分类情感模型，按照上文所述训练模型，将训练好的模型保持不变，不参与后续训练过程。在情感转换模型训练过程中，由于没有真实的并行数据，模型采样无监督方式进行训练，使用下面五种损失约束提升模型生成文本的能力。

（1）**重建损失**：与3.6节中描述的内容相同，重建损失为根据原始文本的中间语义表示以及原始文本的情感值，还原原始文本的损失，表示为。

（2）**反向重建损失**：与3.6节中描述的内容相同，反向重建损失为根据生成文本以及原始文本的情感值，还原原始文本的损失，表示为。

（3）**分类器损失**：与3.6节中描述的内容相同，分类器损失为分类器预测生成文本的预测值，与目标情感值之间的差异，表示为。

（4）**参照损失**：与3.6节中描述的内容相同，参照损失为生成文本与目标情感值对应伪平行句子之间的差异，表示为。

（5）**语言模型损失**：本文参照Yang等人（2018）的想法引入了基于双向GRU的语言模型，该模型在所有训练实例上进行了预训练，以确保本文的模型可以生成流畅的输出句子。对于双向语言模型的正向，通过最小化模型的预测概率分布与正向语言模型的预测概率分布的交叉熵来减小它们的分布差异，计算方式如公式4.7所示：

|  |  |  |
| --- | --- | --- |
|  |  | (4.7) |

其中和分别是情感转换模型的预测概率分布和正向语言模型的预测概率分布，在每个时间步，将情感转换模型的输出单词的连续逼近，即单词嵌入的加权总和与当前概率向量作为语言模型的输入。对于双向语言模型的反向，我们将输出句子反转输入到语言模型中，并以同样的方式计算语言模型的损失。最后，总的语言模型损失定义为前向语言模型损失和后向语言模型损失的平均值。

模型训练的目标是最小化五个损失函数的加权和，如公式4.8所示：

|  |  |  |
| --- | --- | --- |
|  |  | (4.8) |

## 4.6 本章小结

本章对提出模型FGSTDP+LM的相关内容进行详细阐述，首先介绍模型的基本定义，并介绍模型的总体框架。接着介绍模型训练前的准备工作，即情感内容提取和使用依存句法方法构造伪平行句子对。然后介绍基于双向GRU的语言模型相关内容，最后介绍FGSTDP+LM模型的模型结构和FGSTDP+LM的训练过程。

# 5 [实验与分析](#_Toc283328287)

本章将在公开数据集上设计对比实验和消融实验，并使用四个评价指标来验证本文提出模型的有效性。本章将会详细介绍实验中使用的设备信息、数据集情况、对比算法和评价指标，并阐述实验设计的具体方案以及实验结果分析，最后对本章进行总结。

## 5.1 实验环境及数据集

### 5.1.1 实验环境

本文中所有实验均在统一的实验环境中进行，使用实验室配置的服务器进行实验，实验在Linux操作系统中运行。实验代码主要由Python语言编写并使用深度学习框架Pytorch构造模型结构，实验结果需要用到Java语言搭建的情感管道进行情感分析，本文的实验环境配置如表5.1所示。

表5.1 实验环境

|  |  |  |
| --- | --- | --- |
| 环境 | 参数 | 配置 |
| 硬件环境 | CPU | Intel(R) Xeon(R) E5-2630 v4 2.20GHz |
| GPU | GeForce GTX 1080 Ti |
| 内存 | 477G |
| 硬盘 | 7.3T |
| 软件环境 | 操作系统 | Ubuntu 5.4.0 |
| Python | Python3.6 |
| Pytorch | Pytorch0.4.0 |
| Java | jdk1.8.0\_211 |
| 开发工具 | JetBrains PyCharm |

### 5.1.2 实验数据

本文中所有实验使用的数据集均为公开数据集Yelp，Yelp数据集为用户评论数据集，每条数据包含用户评论及该评论的评分，评分在1~5范围内。本文使用Liao等人[41]处理之后的Yelp数据集，经过清理之后，获得了约52万条数据，并从原始数据中评分为3的句子中随机选择8万个句子作为中性句子添加进去。为了确保添加的句子描述的是同一领域，仅选择句子所有词汇都位于原始数据词汇表中的句子，原始数据集的词汇量为9,625。最后得到的数据集总共包含大约60万条句子，随机分配5万条数据作为测试集，1万条数据作为验证集，剩余数据作为训练集。在模型训练阶段，每个输入句子需要与一个情感值相关联。在测试阶段，需要测量生成句子的情感值，以检测生成的句子是否满足指定的目标情感值。因此需要一种自动的方法来测量训练句子和生成句子的情感值，为此Liao等人[41]使用了Stanford CoreNLP中的情感分析器。具体来说，首先调用CoreNLP分析器预测句子在1、2、3、4、5情感值上的概率分布，然后将概率乘以情感值的总和作为句子的最终情感值。表5.2给出了数据的分布情况。之所以没有使用Yelp用户给出的原始评分，有两个原因：（1）希望训练句子的评分和生成句子的评分采用一致的评估方法；（2）实际上可以发现CoreNLP的预测值与用户给出的情感值具有0.85的皮尔逊相关性。原始Yelp数据中仅仅只要整个评论具有评分，而此处使用包含该句子的所有评论的情感值的平均值作为最终情感值。

表5.2 数据集分布

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 情感区间 |  |  |  |  |
| 数据量 | 34576 | 233916 | 166566 | 169196 |

## 5.2 实验评价指标

文本情感极性转换任务有很多评估指标，这些指标也可以用于细粒度情感转换任务。由于缺乏并行语料库以及情感值的细粒度化，本文为任务选择了合适的指标，如下所示。

（1）BLEU（Bilingual Evaluation Understudy）：BLEU[56]最初用于机器翻译任务，用来测量翻译文本和参考文本之间的相似性。BLEU的值从0到1，取值越靠近1表示翻译结果越好，本文参考以前研究工作将其扩展到0到100。随着文本风格转换的出现，BLEU也经常被用于此任务以评估生成句子的内容完整性。由于没有参考文本，因此计算原始文本和生成文本之间的BLEU值。BLEU的计算方式如公式5.1~5.2所示：

|  |  |  |
| --- | --- | --- |
|  |  | (5.1) |
|  |  | (5.2) |

其中（Brevity Penalty）为“过短惩罚”，为生成句子的长度，为原始句子的长度，的取值范围为，生成句子越短，的值越接近0。表示权重，为n-gram精确度，本文计算4-gram内的精确度。

（2）MED（Minimum Edit Distance）：俄罗斯科学家Vladimir Levenshtein[57]在1965年提出了编辑距离度量指标，因此该指标也被命名为Levenshtein Distance。编辑距离常常用于测量两个序列的相似性，通常编辑距离是指将一个单词变换到另一个单词所需的最小单字符操作次数。本文将两个句子和的编辑距离表示为，其中和分别为和的句子长度，编辑距离的计算方式如公式5.3所示：

|  |  |  |
| --- | --- | --- |
|  |  | (5.3) |

（3）MAE（Mean Absolute Error）：由于本文模型的任务是修改句子，使其情感值（由Stanford CoreNLP预测）满足指定的目标情感值，因此将MAE定义为目标情感值与修改句子的情感值之间的平均绝对误差，以此来衡量模型转换文本情感的准确性，MAE值的计算方式如公式5.4所示：

|  |  |  |
| --- | --- | --- |
|  |  | (5.4) |

（4）Fluency：为了测量生成句子的语言流畅度，我们计算每个生成序列的困惑度[55]（PPL），使用4.3中预先训练好的语言模型来测量。计算方式如公式5.5所示：

|  |  |  |
| --- | --- | --- |
|  |  | (5.5) |

该式可以分解为条件概率项的乘积，为了估计条件概率，使用N-gram（Brown等[58]，1992）或神经语言模型进行计算。在本文的实验中，使用KenLM（Heafield[59]，2011年）和Europarl单语英语语料库（Koehn[60]，2005年）训练了4-gram的语言模型，该模型包含混合域文本。

## 5.3 实验对比算法

本文设计了两种对比实验，分别为情感极性转换对比实验和细粒度情感转换对比实验，在两种任务中证明本文提出的FGSTDP和FGSTDP+LM模型的有效性。本文将5种最新模型与两个模型进行比较，前一种模型专门为细粒度情感转换对比实验设计，全部模型将参与情感极性转换对比实验。下面将简要介绍这5种模型：

（1）Sequence Editing under Quantifiable Guidance（QuaSE[41]）：QuaSE首先提出了量化情感转换任务，使用两个编码器捕获语义内容和情感内容，一个解码器生成满足要求的文本。为了更好地分开这两个因素，QuaSE使用伪平行语句来增强模型解开语义内容和情感内容的能力。在测试阶段，QuaSE假设输入文本的情感内容及对应情感值遵循一个高斯分布，然后根据目标情感值选择分布中的最佳内容传递给解码器以改写解开后的语义内容。

（2）Text Style Transfer by Cross-Alignment（TCA[23]）：TCA将输入语句映射到与风格无关的内容表示形式，并将其传递给与风格有关的解码器。它采用对齐自动编码器代替典型的变分自动编码器，通过交叉对齐方式获得两个分布式约束，并使用两个鉴别器来修改句子。

（3）Delete, Retrieve, Generate: A Simple Approach to Sentiment and Style Transfer（Delete[32]）：Delete提出了一种更简单的方法，由于观察到文本风格通常有独特的短语标记（例如“too small”）。Delete通过删除与句子的原始风格相关的短语来提取语义内容，并检索和目标风格密切相关的新语句，然后使用神经模型将这些元素流畅地组合为最终输出。Delete提出了4种方法：DeleteOnly、DeleteAndRetrieve、TemplateBased和RetrieveOnly，并证明了DeleteAndRetrieve是最有效的方法，因此本文使用DeleteAndRetrieve方法作为对比模型。

（4）Unpaired Sentiment-to-Sentiment Translation: A Cycled Reinforcement Learning Approach（Unpaired[35]）：Unpaired提出了一种循环强化学习方法，该方法可以通过中和模块和情感化模块之间的协作来对非平行的数据进行训练。中和模块负责过滤掉情感词来提取非情感语义信息，情感化模块负责将情感添加到已中和的语义内容中，以进行情感到情感的转换。

（5）A Dual Reinforcement Learning Framework for Unsupervised Text Style Transfer（Dual[61]）：Dual提出了一种双重强化学习框架，可通过一个一步映射模型直接转换文本的风格，而不会对内容和风格进行分离。具体地，将学习源风格到目标风格和目标风格到源风格映射视为双重任务，并基于这种双重结构设计了两个奖励机制，分别反映了风格转换的准确性和内容完整性。这样就可以通过强化学习来训练两个单步映射模型，而无需使用任何并行数据。

## 5.4 实验设置与方案设计

### 5.4.1 实验参数设置

对于情感极性转换和细粒度情感转换任务，FGSTDP和FGSTDP+LM模型使用的编码器是具有500维隐藏状态的单层双向GRU。解码器是含有注意力机制的单层双向GRU，其隐藏状态的维数设置为1000维。编码器的输出，即编码之后的中间语义表示，与目标情感嵌入相结合，作为解码器的输入。情感嵌入的维度为128维。编码器-解码器结构训练的学习率为0.0005。分类器由隐藏层大小为200的编码器（GRU）和隐藏层大小为100的全连接层（MLP）组成，分类器的学习率为0.001。语言模型的嵌入层大小为200，隐藏层大小为500，训练时的学习率为0.001。情感值差异和相似性的权重分别设置为1和0.5。对于5个损失函数的权重（、、、、），根据验证集使用不同的值对其进行调整，最后将它们分别设置为0.7、0.2、0.3、0.7、0.5。对于对比算法，本文遵循其源代码中的默认参数。

### 5.4.2 实验方案设计

实验方案主要分为四部分，第一部分主要验证本文提出的两个模型在细粒度情感转换任务中的有效性，该部分使用5.3节中描述的QuaSE算法与本文提出的模型进行对比，并将FGSTDP和FGSTDP+LM模型进行对比，使用5.2节中描述的BLEU、MED、MAE和Fluency四个指标对结果进行衡量。第二部分主要验证两个模型在情感极性转换任务中的有效性，这部分使用5.3节中描述的全部算法与本文提出的模型进行对比，使用5.2节中描述的BLEU和转换准确度Accuracy指标对结果进行分析。第三部分设计了消融实验，分别对模型的每一部分，即通过对4.5节提出的四个损失函数（重建损失、反向重建损失、参照损失、语言模型损失）进行验证，证明每个部分的合理性和有效性。

## 5.5 实验结果及分析

### 5.5.1 FGTST和FGSTDP+LM模型在细粒度情感转换中的有效性分析

在细粒度情感转换实验中，本文将FGSTDP和FGSTDP+LM模型与QuaSE、FGST模型进行比较。每个输入句子都需要转换为五个情感值分别满足目标值1、2、3、4、5的句子。在目标情感和生成句子的情感强度之间进行MAE指标评估，并评估生成句子和输入句子之间的编辑距离MED和BLEU值，以及生成句子的流畅性Frequency。BLEU和MED值反映生成句子的内容完整性，结果分别显示在表5.3和表5.4。MAE值反馈生成句子的准确性，结果显示在表5.5。Frequency值反馈生成句子的流畅性及可读性，结果显示在表5.6，表格中所有数值均为在整个数据集上结果的平均值。“original”是指使用原始输入来计算评估指标，由于其与自身求BLEU和MED没有意义，因此只计算原始输入的情感值和目标情感值的MAE以及其可读性，“original”的结果可以反映模型是否达到要求。

由表5.3所示，FGSTDP模型在5个情感值上的BLEU值比QuaSE分别提高了12.66、6.18、1.33、1.65、18.19，平均提高了8.00。这表明FGSTDP模型比QuaSE模型在内容完整性上表现更好。主要原因是因为QuaSE模型分别学习了与输入句子分开的内容和情感表示，但实际上很难将它们完全分开，并且会导致部分内容丢失。但是，FGSTDP模型不学习解开的表示，而是使用一些约束条件使内容保持不变。而FGSTDP+LM模型在5个情感值上的BLEU值比FGSTDP模型分别提高了7.47、8.59、4.51、-3.51、3.16，平均提高了2.55。总体来说FGSTDP+LM模型在BLEU值上略优于FGSTDP模型，因为FGSTDP+LM模型使用了语言模型来优化生成句子的流畅性，改善了FGSTDP模型生成句子词汇重复、可读性较差的缺陷，使得生成句子更接近原始句子的语义。

表5.3 模型生成句子的BLEU值

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 模型 | T=1 | T=2 | T=3 | T=4 | T=5 |
| Original | N/A | N/A | N/A | N/A | N/A |
| QuaSE | 6.2635 | 24.5512 | 24.6344 | 30.2176 | 8.2306 |
| FGSTDP | 18.9311 | 30.7360 | 25.9682 | **31.8652** | 26.4239 |
| FGSTDP+LM | **26.3982** | **39.3276** | **30.4792** | 28.3511 | **29.5823** |

为了全方位衡量生成句子的内容完整性，本文还使用了编辑距离指标，该指标的结果见表5.4。如表中数据所示，FGSTDP模型的编辑距离在5个情感值上比QuaSE都小，分别降低了4.99、2.21、2.06、2.63、4.33，平均降低了3.25，这也显示了FGSTDP模型比QuaSE模型的内容保留完整性更好。而FGSTDP+LM模型的编辑距离比FGSTDP模型在5个情感值上分别降低了1.56、2.47、1.04、0.04、2.64，平均降低了1.56，这也表明FGSTDP+LM模型使用语言模型的有效性，可以在一定程度上提升句子的语义完整性。

表5.4 模型生成句子的MED值

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 模型 | T=1 | T=2 | T=3 | T=4 | T=5 |
| Original | N/A | N/A | N/A | N/A | N/A |
| QuaSE | 11.8824 | 8.7813 | 8.3619 | 8.1327 | 11.5809 |
| FGSTDP | 6.8879 | 6.5622 | 6.3015 | 5.4971 | 7.2421 |
| FGSTDP+LM | **5.3211** | **4.0831** | **5.2539** | **5.4519** | **4.6010** |

在计算MAE评估指标之前，为了不让句子中重复的情感词影响情感值预测，本文将生成句子中重复的情感词去掉之后再来计算MAE值，得到表5.5中的结果。在该表中，FGSTDP、FGSTDP+LM和QuaSE模型在5个情感值上的MAE值均小于“original”，这表明这些模型都有能力修改文本的情感。此外，FGSTDP模型在5个情感值上的MAE值比QuaSE模型分别降低了34.45%、-59.48%、16.76%、34.14%、18.83%，平均降低了8.94%，总体来说FGSTDP模型在情感转换上更准确。主要原因是FGSTDP模型利用伪平行句子与生成句子之间的误差以及分类器误差为解码器提供了更有效和更丰富的反馈，指导模型更好地生成满足目标情感的句子。相比之下，QuaSE假设情感内容服从一个高斯分布，通过采样来进行情感修改，这并不是那么精确。而FGSTDP+LM模型在5个情感值上比FGSTDP模型分别降低了4.29%、2.40%、32.89%、-56.95%、24.07%，平均降低了1.34%，这表明提高生成句子的流畅性可以在一定程度提升转换准确性。

表5.5 模型生成句子的MAE值

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 模型 | T=1 | T=2 | T=3 | T=4 | T=5 |
| Original | 2.1376 | 1.1502 | 0.8136 | 1.0013 | 1.8721 |
| QuaSE | 1.2985 | **0.5773** | 0.7734 | 0.6710 | 1.1954 |
| FGSTDP | 0.8512 | 0.9207 | 0.6438 | **0.4419** | 0.9703 |
| FGSTDP+LM | **0.8146** | 0.8986 | **0.4320** | 0.6936 | **0.7367** |

除了衡量生成句子的内容完整性和情感转换准确性，本文还对生成句子的语言流畅性进行评估，以此验证FGSTDP+LM模型提出的合理性。结果显示在表5.6中，如表中数据所示，FGSTDP模型在2、3、4情感值上的语言流畅性比QuaSE差，分别上升了0.23、0.33、0.69，虽然在1和5情感值上表现更好，但这是因为QuaSE模型在两端极值上发挥不稳定的原因。由于FGSTDP模型生成的句子在语言流畅性方面表现较差，因此FGSTDP+LM模型引入语言模型来改善FGSTDP模型句子流畅性的问题。由表4.6中的数据可以看到，FGSTDP+LM模型在5个情感值上的语言流畅性均比QuaSE模型好，分别降低了0.05、0.95、0.53、1.17、0.26，平均降低了0.59，这验证了语言模型对提升文本流畅性的有效性。

表5.6 模型生成句子的Frequency值

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 模型 | T=1 | T=2 | T=3 | T=4 | T=5 |
| Original | N/A | N/A | N/A | N/A | N/A |
| QuaSE | 8.1867 | 3.7319 | 3.5028 | 3.4325 | 6.3045 |
| FGSTDP | 3.6643 | 3.9627 | 3.8390 | 4.1302 | 3.8319 |
| FGSTDP+LM | **3.6138** | **3.0055** | **3.3068** | **2.9599** | **3.5697** |

为了直接展示本文提出模型在细粒度情感转换的有效性，表5.7中给出了FGSTDP和FGSTDP+LM模型以及QuaSE生成句子的一些示例。每个句子被修改为5个句子，它们的情感值分别为1、2、3、4、5。在示例中，由于输入文本的情感标签为5，因此在T=5上生成的句子与输入句子相同。在表中可以看到输入句子描述的语义内容主要为“music”，QuaSE模型的输出与原始语义相差甚远，并且输出句子的可读性较差，与目标情感值的差距也较大。而FGSTDP模型的输出与原始语义内容相同，都是在描述“music”，在目标情感值为1时将输入句子的情感内容“fun”变换为“waste”，实现了情感的转换。在目标情感值为2、3、4时，情感差异较大并且出现词汇重复、语义不通的问题。为了解决FGSTDP模型的不足，FGSTDP+LM模型提出了使用语言模型。该模型在目标情感值为1时将输入句子的情感内容“fun”变换为“waste, do not”；在目标情感值为2时，将其变换为“not…fun”；在目前情感值为3时将其去掉；在目标情感值为4时将其变换为“always ok”。这四种情况下都达到了情感转换的要求，并且句子语义流畅、可读性较强，由此验证了FGSTDP+LM模型的有效性。

表5.7 生成句子示例

|  |  |
| --- | --- |
| Input | the live music is fun as well! (情感值为5) |
| Output | QuaSE |
| T=1  T=2  T=3  T=4  T=5 | no no lack worse no it unprofessional it n't bland not no it not no.  am am am am am am that this at the buffett its to seem the.  good portions served in a casual beer.  the start now seven sides flavors & ages wonderful flavors with red spot-on, red  the live music is fun as well! |
| Output | FGSTDP |
| T=1  T=2  T=3  T=4  T=5 | the music music is **waste** as well.  the live music is music authentic as those park as clean.  the live music is fun as well!  the live music chops great gifts chips.  the live music is fun as well! |
| Output | FGSTDP+LM |
| T=1  T=2  T=3  T=4  T=5 | the music is **waste**, **do not**.  the live music is **not as fun**.  the live music is.  the live music is **always okay**.  the live music is fun as well! |

### 5.5.2 FGTST和FGSTDP+LM模型在情感极性转换中的有效性

在情感极性转换实验中，将5.3节中的所有模型作为此实验的对比模型，使用情感转换准确率作为衡量指标，以验证本文提出模型在情感极性转换中的有效性。如第3节所述，本文定义情感值大于3的句子为积极文本，小于3的句子为消极文本，情感极性转换的目标是将积极文本转为消极文本，并保留原始文本的语义内容，反之亦然。实验结果显示在表5.8中，本文从两个方向的准确率来反应模型情感转换的准确性。从表5.8中可以看到，所有模型将消极转积极上的准确率高于积极转消极，这是因为在训练数据中积极文本比消极文本多的原因。在消极转积极方向上，QuasSE模型的准确率最高，其次依次是FGSTDP+LM模型、Delete模型、FGSTDP模型、Dual模型、Unpaired模型、TCA模型，虽然在这个方向上本文提出的模型不是最好的，但效果可以跟QuaSE模型相媲美。在积极转消极方向上，FGSTDP+LM模型的准确率最高，其次是FGSTDP模型。FGSTDP和FGSTDP+LM模型在两个方向上的准确率的平均值高于其它所有模型，这验证了本文提出的两个模型在情感极性转换上的有效性。

表5.8 情感极性转换实验结果

|  |  |  |  |
| --- | --- | --- | --- |
|  | Neg. to Pos. | Pos. to Neg. | Avg. acc |
| *QuaSE* | **89.81%** | 76.93% | 83.36% |
| *TCA* | 73.80% | 69.12% | 71.46% |
| *Delete* | 85.65% | 82.61% | 84.13% |
| *Unpaired* | 79.17% | 73.52% | 76.34% |
| *Dual* | 79.40% | 75.00% | 77.20% |
| *FGSTDP* | 85.37% | 83.36% | 84.36% |
| *FGSTDP*+*LM* | 87.18% | **84.76%** | **85.97%** |

### 5.5.3 消融实验

在本文中，通过引入了一个分类器和4.5节中提出的四个约束来指导编码器-解码器修改句子。为了验证这四种损失函数的有效性，本文在MAE和BLEU指标下进行了消融研究。分别去除这四项损失函数，同时保持其他三项损失函数不变，来验证去除的那个损失函数的有效性。实验结果显示在表5.9和表5.10中，每个表中的第一行是情感强度值，在实验中仅考虑情感值1、3和5。每个表中的第二行为对比模型QuaSE的MAE和BLEU值，用于比较。表格中接下来的五行显示了在去掉所有损失、去掉参考损失、重建损失、反向重建损失和语言模型损失情况下的MAE和BLEU值，最后一行显示保留所有损失的MAE和BLEU值。

在表5.9中，“None”的BLEU值的平均值比QuaSE小，说明分类器对内容完整性的贡献不大。与“None”相比，去掉每个损失的BLEU值的平均值分别提升了160.70%、135.55%、127.19%、149.84%，说明任意三种损失函数的组合都有效地指导模型保留原始语义内容。与“ALL”相比，去掉每个损失的BLEU值的平均值分别降低了10.95％、17.92%、22.57％、12.45%，结果表明去掉每种损失函数都会使得语义内容的丢失，因此每种损失函数都有助于提高语义内容完整性。而四种损失函数对内容完整性的贡献不同，由高到底分别是参考损失、反向重建损失、语言模型损失、重建损失。

表5.9消融实验的BLEU结果

|  |  |  |  |
| --- | --- | --- | --- |
|  | T=1 | T=3 | T=5 |
| *QuaSE* | 6.26 | 24.63 | 8.23 |
| *None* | 8.11 | 10.03 | 12.32 |
|  | 24.90 | 28.79 | 23.17 |
|  | 18.93 | 25.96 | 26.42 |
|  | 20.65 | 23.34 | 22.91 |
|  | 23.09 | 21.19 | 31.24 |
| ALL | 26.39 | 30.48 | 29.58 |

根据表5.10中的数据可见，“None”的所有MAE值小于表5.5中的“Original”的MAE值，并且“None”的MAE值的平均值小于“QuaSE”，这表明模型通过分类器的指导可以修改句子的情感强度。与“None”相比，消除每个损失的MAE值的平均值分别降低了13.50％、2.65％、-0.32％和9.55%，这表明任意三种损失函数的组合都有利于情感转换准确性。与“ALL”相比，消除每个损失的MAE值的平均值升高幅度分别为15.80％，29.36％、36.97％和20.07%。这些表明，每一个损失函数都对情感的修改做出了一定的贡献，贡献大小由高到低分别是参考损失、反向重建损失、语言模型损失、重建损失。此外，在所有情感值中，“ALL”的MAE和BLEU值的平均值是最佳的，由此表明这四种损失的组合有效地增强了模型修改情感以及保留原始语义内容的能力。

表5.10消融实验的MAE结果

|  |  |  |  |
| --- | --- | --- | --- |
|  | T=1 | T=3 | T=5 |
| *QuaSE* | 1.29 | 0.77 | 1.19 |
| *None* | 1.38 | 0.53 | 1.09 |
|  | 0.92 | 0.56 | 0.95 |
|  | 0.90 | 0.75 | 0.93 |
|  | 0.94 | 0.78 | 1.03 |
|  | 0.85 | 0.64 | 0.97 |
| ALL | 0.83 | 0.42 | 0.92 |

## 5.6 本章小结

本章对提出的FGSTDP和FGSTDP+LM模型设计实验，包括对比实验和消融实验，以验证本文提出模型的有效性和模型各个部分的合理性。本章首先介绍实验环境及数据集情况，然后介绍实验结果的评价指标，包括BLEU、MED、MAE和Fluency。接着介绍了实验中用到的对比算法，并对实验参数设置和方案设计进行了详细描述。最后将FGSTDP和FGSTD+LM模型与对比算法在细粒度情感转换和情感极性转换实验中的结果进行展示，并对最终结果进行分析证明模型的有效性，使用消融实验验证模型每个部分的合理性。

# [6 总结与展望](#_Toc283328293)

本章首先对本文研究工作及取得的成果进行总结，进一步讨论工作展望，介绍未来可以在本文工作基础上进行扩展及优化的研究方向。

## 6.1 论文总结

文本风格转换作为自然语言处理的热门研究方向，拥有广泛的实际应用。文本风格转换在文本原有风格上进行风格变换并保留原始语义内容，由于其拥有较高的研究价值，引起了无数研究者的注意，许多有效的模型和算法应运而生，取得了不错的研究成果。文本风格多种多样，涉及情感、时态语态等风格，其中文本情感被研究者们广泛研究，提出了基于强化学习、基于记忆网络的文本情感转换模型等。

现有对文本情感转换的研究主要集中在积极和消极二元情感转换，对细粒度文本情感转换这项特定的任务关注度不足，难以应用在复杂的文本情感转换任务上。并且现有的方法大多会将文本的语义和情感这两个因素分开，然而这两个因素互相影响，很难完全分开，分开之后会导致信息内容的丢失。而且大部分现有的工作会根据情感来构建多个解码器，不同的解码器对应不同的情感，这种方法相对没有那么灵活，对于细粒度情感转换十分不友好。针对现有方法的不足，本文提出了基于依存句法分析的细粒度情感转换模型和基于语言模型的细粒度情感转换模型，具体的研究工作如下：

(1) 使用依存句法分析对文本词汇间的依存关系进行分析，提取出情感词汇与其修饰的上下文对象。通过使用提取出来的情感词汇与特定上下文对象的关联关系来构造伪平行句子对，为模型提供训练信号，将无监督训练进一步转换为伪监督训练。

(2) 对于文本情感转换任务，现有的方法倾向于将文本的情感和语义内容分开，并在分开之后的语义内容上进行文本修改，这种方法会使得编码之后的中间表示损失部分语义信息。本文针对细粒度文本情感转换，提出了FGSTDP模型，该模型结合依存句法分析生成目标情感文本，并直接在原始文本上进行修改，不对情感和语义内容进行分离。在此模型基础上，为了提高生成句子的可读性，FGSTDP+LM模型引入了预训练的语言模型。

(3) 为了验证FGSTDP和FGSTDP+LM模型的有效性，将模型在公开数据集上与对比方法进行比较，使用BLEU、MAE、MED、Perplexity四个评价指标进行评估，并且设计消融实验验证模型结构每一部分的合理性与有效性。

## 6.2 论文展望

本文结合依存句法分析，利用情感词汇与修饰对象的特定依存关系，构造伪平行句子对，并使用多元分类器以及重建损失、反向重建损失等约束条件来指导模型完成细粒度情感转换任务，并且还使用预训练语言模型改善生成句子的可读性。虽然本文提出的两个模型取得了不错的研究成果，但仍然存在一些不足以及可进一步优化的地方，主要有以下几个方面：

(1) 研究更多的文本风格属性。本文研究的文本风格主要为情感，但文本还拥有很多其它的风格属性，比如文本正式化非正式化、文本的时态、文本的语态、文本的书写风格等属性。可以研究将模型通用化，不仅适用单一的风格属性，还可以转换文本的任一风格。

(2) 研究文本多风格属性转换。现有的文本风格转换方法在一个训练阶段只能转换文本的一个属性，对于多个风格属性需要重新训练模型来完成对不同的风格的转换，对于多风格属性而言，这种操作非常耗时。因此可以考虑同时转换文本的不同属性，比如同时转换非正式的拥有现在进行时态的文本为正式化的拥有过去时态的文本。这种方案可以节省模型训练的时间，也使得模型功能更强大更灵活。

(3) 研究长文本风格转换。现有的文本风格转换研究主要针对短文本，因为短文本序列的语法结构更简单，序列词汇间的依赖关系更容易捕捉。并且现有的深度学习模型比如RNN、LSTM、GRU等只能处理短期依赖关系，对于长文本序列而言显得很吃力，无法捕捉较长序列词汇间的依赖关系。针对深度学习模型无法处理长文本序列的问题，Transformer、Bert等自编码语言模型被提出，这些模型很好地捕捉了长文本的依赖关系，因此可以考虑使用这些模型来尝试对长文本的风格进行转换。

(4) 研究表达含蓄的文本进行转换。现有的文本风格转换研究对清晰表达风格的文本（比如“The food is delicious”）比较友好，对含蓄表达的文本（比如“This thing has been in the closet for a long time and it has accumulated dust”）效果较差。主要是模型对文本的理解不够，因此可以研究对含蓄表达的文本进行转换，使模型泛化性能更好。

# 致谢

时光如白驹过隙，三年研究生生涯即将结束，想起拉着行李箱第一次站在华中科技大学校门口石碑边上，望着高大庄严的毛主席像心情无比激动，对新的学习阶段充满憧憬与忐忑，回首往事恍若昨日。在华中科技大学的三年学习与生活中，我感受到了浓厚的学习氛围与积极向上的正能量。华科学子们勤奋学习，教室实验室中总是能看到学生们忙碌的身影，校园每天的定时广播播报着新闻、校园资讯等，操场上学生们挥洒着运动的汗水，使得学校充满青春向上的气息，很荣幸能在华中科技大学渡过这三年。在智能与分布计算实验室的学习和科研中，我收获颇丰，实验室学术氛围良好，老师们兢兢业业地为我们传道解惑，关心我们的学习和生活情况，同学们互相学习交流，互帮互助，给我家的温暖。在此我要感谢所有帮助过我的人，感谢他们为我的研究生生涯画上一个圆满的句号。

首先我要感谢我的导师李瑞轩，感谢李老师三年前保研的时候给我进入智能与分布计算实验室学习的机会。在入学前，李老师对我本科的毕设论文进行指导，帮助我选题，提供教程和学习视频，让我提前接触相关基础知识，为未来的研究学习做铺垫。李老师学识渊博，做事认真负责，对每个学生都关怀备至，经常与学生进行沟通交流，了解每个学生的学习生活情况。在科研上，李老师帮助我确定研究方向，组织开展每周至少一次的学术讨论和组会，促进同学们的学术交流，了解我们的研究进度以及指导我们的研究内容，在每次学术讨论中我都有所收获，学习到了很多有用的知识。在投递会议论文时，李老师给了我最大的支持，帮助我确定合适的会议，指导我的工作与论文撰写，我的论文能被会议录取发表离不开李老师的悉心指导。在生活中，李老师为人谦逊随和，注重我们的全面发展，组织实验室每周一次的羽毛球活动，并积极参与。支持我们去公司实习，提前适应企业工作节奏，因此我们毕业后大部分都能找到很好的工作。三年的研究生学习和生活得以圆满，离不开李老师的指导与关怀，因此由衷地感谢李老师。

感谢电子与通信学院的周潘老师和瞿晓晔学长。瞿晓晔学长在我定了大的研究方向之后与我沟通交流，帮我确定了进一步研究的具体内容，让我阅读相关论文并每周与我讨论论文的内容以及想法，在论文撰写以及科研方法上给了我很多意见，帮助我的论文顺利投递出去。周潘老师在每周的报告上，关心我的研究进展并提出相应的意见与看法。还有感谢实验室的李玉华老师和辜希武老师在我研究生阶段给予的帮助与支持。

感谢实验室同级的同学们，虽然很多同学已经毕业了，但他们这三年来给予了我很多帮助与陪伴。感谢刘洋同学在本科毕设论文中给予的帮助与意见，还有在入学之后跟我分享学术心得以及学习方法，带我一起参加学术比赛，在我找实习和工作的时候将经验毫无保留地分享给我。感谢孔令晓和张瑜同学这三年来的陪伴，在生活和学习中给予了我很多帮助与关心，让我三年的研究生生活过得很充实和温暖。感谢实验室的师兄师姐们一直以来的关心与帮助，以及师弟师妹们的陪伴，让实验室组成了一个温暖的大家庭。还有感谢我的室友们和朋友们，我们一起谈天说笑，分享生活趣事，在难过时给予安慰，在快乐时分享喜悦，就像一家人一样，很庆幸能认识你们。

最后我想把最真挚的感谢留给我的家人。感谢我的父亲和母亲在我的每个人生阶段对我的支持与无私奉献，在家庭困难时候全力支持我读研的梦想，在我生病时候的担心与牵挂，包容我的不懂事与任性，在外人面前表现出对我的骄傲和自豪。我的父母都是普通人，但他们任劳任怨，尽他们所能给我最好的生活，给予我无微不至的关心与关爱。感谢我的男朋友这三年来的陪伴与支持，以及对我的包容和帮助，异地恋不易但我们都一起过来了，未来我们还将继续同行。

肖露露

二零二一年五月

# 参考文献

1. Gatys, Leon A., Alexander S. Ecker, et al. Image style transfer using convolutional neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR). Las Vegas, USA, 27-30 June 2016, IEEE Computer Society 2016: 2414-2423.
2. Yufei Wang, Zhe Lin, Xiaohui Shen, et al. Skeleton Key: Image captioning by skeleton-attribute decomposition. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR). Honolulu, USA, 21-26 July 2017, IEEE Computer Society 2017: 7272-7281.
3. 张惊雷, 厚雅伟. 基于改进循环生成式对抗网络的图像风格迁移. 电子与信息学报. 2020, 42(5): 1216-1222.
4. Igor Melnyk, Cícero Nogueira dos Santos, Kahini Wadhawan, et al. Improved neural text attribute transfer with non-parallel data. arXiv preprint arXiv: 1711.09395.
5. Shrimai Prabhumoye, Yulia Tsvetkov, Ruslan Salakhutdinov, et al. Style Transfer Through Back-Translation. In: Annual Meeting of the Association for Computational Linguistics (ACL). Melbourne, Australia, 15-20 July 2018, Association for Computational Linguistics, 2018: 866-876.
6. Diederik P. Kingma, Max Welling. Auto-encoding variational bayes. In: International Conference on Learning Representations (ICLR). Banff, Canada, 14-16 April 2014, Conference Track Proceedings 2014: 1312.6114.
7. Jiwei Li, Will Monroe, Tianlin Shi, et al. Adversarial Learning for Neural Dialogue Generation. In: The Conference on Empirical Methods in Natural Language Processing (EMNLP). Copenhagen, Denmark, 9-11 September 2017, Association for Computational Linguistics 2017: 2157-2169.
8. Blei D M, Kucukelbir A, McAuliffe J D. Variational inference: A review for statisticians. Journal of the American statistical Association. 2017, 112(518): 859-877.
9. Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, et al. Generating Sentences from a Continuous Space. In: The Conference on Computational Natural Language Learning (CoNLL). Berlin, Germany, 11-12 August 2016, ACL 2016: 10-21.
10. Fu Z, Tan X, Peng N, et al. Style transfer in text: Exploration and evaluation. In: Proceedings of the AAAI Conference on Artificial Intelligence. New Orleans, USA, 2-7 February 2018, AAAI Press 2018: 663-670.
11. Yifan Gao, Jianan Wang, Lidong Bing, et al. Difficulty Controllable Question Generation for Reading Comprehension. arXiv preprint arXiv: 1807.03586.
12. 聂锦燃, 魏蛟龙, 唐祖平. 基于变分自编码器的无监督文本风格转换. 中文信息学报, 2020, 34(7): 79-88.
13. 赵丽飞, 余航, 谢清涛,等. 基于卷积神经网络的梵高作品风格转换. 电子技术与软件工程. 2019, 156(10): 70.
14. Jain P, Mishra A, Azad A P, et al. Unsupervised controllable text formalization. In: Proceedings of the AAAI Conference on Artificial Intelligence. Hawaii, USA, 27 January - 1 February 2019, AAAI Press 2019, 33(01): 6554-6561.
15. Yunli Wang, Yu Wu, Lili Mou, et al. Formality Style Transfer with Shared Latent Space. In: Proceedings of the 28th International Conference on Computational Linguistics. Barcelona, Spain(Online), 8-13 December 2020, International Committee on Computer Linguistics 2020: 2236-2249.
16. Subramanian S, Lample G, Smith E M, et al. Multiple-attribute text style transfer. arXiv preprint arXiv: 1811.00552, 2018.
17. Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial networks. Communications of the ACM. 2020, 63(11): 139-144.
18. Kingma, Diederik P, Max Welling. Auto-encoding variational Bayes. arXiv preprint arXiv:1312.6114, 2013.
19. Zichao Yang, Zhiting Hu, Chris Dyer, et al. Unsupervised Text Style Transfer using Language Models as Discriminators. In: Conference and Workshop on Neural Information Processing Systems (NeurIPS). Montréal, Canada, 3-8 December 2018, NeurIps 2018: 1805.11749.
20. Zhiting Hu, Zichao Yang, Xiaodan Liang, et al. Toward controlled generation of text. In: International Conference on Machine Learning (ICML). Sydney, Australia, 6-11 August 2017, Proceedings of Machine Learning Researh 2017: 1587-1596.
21. Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, et al. Generating Sentences from a Continuous Space. In: The Conference on Computational Natural Language Learning (CoNLL). Berlin, Germany, 11-12 August 2016, ACL 2016: 10-21.
22. Zhiting Hu, Zichao Yang, Salakhutdinov Ruslan, et al. On unifying deep generative models. In: International Conference on Learning Representations (ICLR). Vancouver, Canada, 30 April – 3 May 2018, Conference Track Proceedings 2018: 1706.00550.
23. Tianxiao Shen, Tao Lei, Barzilay Regina, et al. Style Transfer from Non-Parallel Text by Cross-Alignment. In: Conference and Workshop on Neural Information Processing Systems (NIPS). Long Beach, CA, USA, 4-9 December 2017, NIPS 2017: 6830-6841.
24. Junbo Jake Zhao, Yoon Kim, Kelly Zhang, et al. Adversarially regularized autoencoders. In: International conference on machine learning ICML. Stockholm, Sweden, 10-15 July 2018, Proceedings of Machine Learning Research 2018: 5902-5911.
25. Makhzani Alireza, Shlens Jonathon, Jaitly Navdeep, et al. Adversarial Autoencoders. arXiv preprint arXiv: 1511.05644, 2015.
26. Zhenxin Fu, Xiaoye Tan, Nanyun Peng, et al. Style transfer in text: Exploration and evaluation. In: Proceedings of the AAAI Conference on Artificial Intelligence. Louisiana, USA, 2-7 February 2018, AAAI Press 2018: 663-670.
27. Sutskever Ilya, Vinyals Oriol, Le Quoc V. Sequence to sequence learning with neural networks. In: Conference and Workshop on Neural Information Processing Systems (NIPS). Quebec, Canada, 8-13 December 2014, NIPS 2014: 3104-3112.
28. Prabhumoye Shrimai, Tsvetkov Yulia, Salakhutdinov Ruslan, et al. Style transfer through back-translation. In: Annual Meeting of the Association for Computational Linguistics (ACL). Melbourne, Australia, 15-20 July 2018, Association for Computational Linguistics 2018: 866-876.
29. Rabinovich E, Mirkin S, Patel R N, et al. Personalized machine translation: Preserving original author traits. In: European Conference on Artificial Life (ECAL). Valencia, Spain, 3-7 April 2017, Association for Computational Linguistics 2017: 1074-1084.
30. Hongyu Gong, Suma Bhat, Lingfei Wu, et al. Reinforcement Learning Based Text Style Transfer without Parallel Training Corpus. In: Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT). Minneapolis, USA, 2-7 June 2019, Association for Computational Linguistics 2019: 3168-3180.
31. Yufang Huang, Wentao Zhu, Deyi Xiong, et al. Cycle-Consistent Adversarial Autoencoders for Unsupervised Text Style Transfer. In: International Conference on Computational Linguistics (COLING). Barcelona, Spain (Online), 8-13 December 2020, International Committee on Computational Linguistics 2020: 2213-2223.
32. Juncen Li, Robin Jia, He He, et al. Delete, Retrieve, Generate: a Simple Approach to Sentiment and Style Transfer. In: Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT). New Orleans, Louisiana, USA, 1-6 June 2018, Association for Computational Linguistics 2018: 1865-1874.
33. Yi Zhang, Jingjing Xu, Pengcheng Yang, et al. Learning Sentiment Memories for Sentiment Modification without Parallel Data. In: The Conference on Empirical Methods in Natural Language Processing (EMNLP). Brussels, Belgium, 31 October-4 November 2018, Association for Computational Linguistics 2018: 1103-1108.
34. Ye Zhang, Nan Ding, Soricut Radu. SHAPED: Shared-Private Encoder-Decoder for Text Style Adaptation. In: Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT). New Orleans, Louisiana, USA, 1-6 June 2018, Association for Computational Linguistics 2018: 1528-1538.
35. Jingjing Xu, Xu Sun, Qi Zeng, et al. Unpaired Sentiment-to Sentiment Translation: A Cycled Reinforcement Learning Approach. In: Annual Meeting of the Association for Computational Linguistics (ACL). Melbourne, Australia, 15-20 July 2018, Association for Computational Linguistics 2018: 979-988.
36. Ashish Vaswani, Noam Shazeer, Niki Parmar, et al. Attention is All you Need. In: Conference and Workshop on Neural Information Processing Systems (NIPS). Long Beach, CA, USA, 4-9 December 2017, NIPS 2017: 5998-6008.
37. Ning Dai, Jianze Liang, Xipeng Qiu, et al. Unpaired Text Style Transfer without Disentangled Latent Representation. In: Annual Meeting of the Association for Computational Linguistics (ACL). Florence, Italy, 28 July-2 August 2019, Association for Computational Linguistics 2019: 5997-6007.
38. Akhilesh Sudhakar, Bhargav Upadhyay, Arjun Maheswaran. Transforming Delete, Retrieve, Generate Approach for Controlled Text Style Transfer. In: the Conference on Empirical Methods in Natural Language Processing (EMNLP) & International Joint Conference on Natural Language Processing (IJCNLP). Hong Kong, China, 3-7 November 2019, Association for Computational Linguistics 2019: 3267-3277.
39. Xing Wu, Tao Zhang, Liangjun Zang, et al. "Mask and Infill": Applying Masked Language Model to Sentiment Transfer. arXiv preprint arXiv:1908.08039, 2019.
40. Jacob Devlin, Ming-Wei Chang, Kenton Lee, et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT). Minneapolis, MN, USA, 2-7 June 2019, Association for Computational Linguistics 2019: 4171-4186.
41. Yi Liao, Lidong Bing, Piji Li, et al. Quase: Sequence editing under quantifiable guidance. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. Brussels, Belgium, 31 October - 4 November 2018, Association for Computational Linguistics 2018: 3855-3864.
42. Fuli Luo, Peng Li, Pengcheng Yang, et al. Towards fine-grained text sentiment transfer. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Florence, Italy, 28 July - 2 August 2019, Association for Computational Linguistics 2019: 2013-2022.
43. Dozat Timothy, Manning Christopher D. Deep biaffine attention for neural dependency parsing. arXiv preprint arXiv. 1611-01734, 2016.
44. Dyer Chris, Ballesteros Miguel, Ling Wang, et at. Transition-Based Dependency Parsing with Stack Long Short-Term Memory. In: Annual Meeting of the Association for Computational Linguistics (ACL). Beijing, China, 26-31 July 2015, The Association for Computer Linguistics 2015: 334-343.
45. Zaremba Wojciech, Sutskever Ilya, Vinyals Oriol. Recurrent neural network regularization. arXiv preprint arXiv:1409.2329, 2014.
46. Hochreiter Sepp, Schmidhuber Jurgen. Long short-term memory. Neural computation. 1997, 9(8): 1735-1780.
47. Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, et al. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In: The Conference on Empirical Methods in Natural Language Processing (EMNLP). Doha, Qatar, 25-29 October 2014, ACL 2014: 1724-1734.
48. Ilya Sutskever, Oriol Vinyals, Quoc V. Le. Sequence to Sequence Learning with Neural Networks. In: Conference and Workshop on Neural Information Processing Systems (NIPS). Montreal, Quebec, Canada, 8-13 December 2014, NIPS 2014: 3104-3112.
49. Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space. In: International Conference on Learning Representations (ICLR). Scottsdale, Arizona, USA, 2-4 May 2013, Workshop Track Proceedings 2013:1301-3781.
50. Lund K, Burgess C. Producing high-dimensional semantic spaces from lexical co-occurrence. Behavior research methods, instruments & computers. 1996, 28(2): 203-208.
51. Deerwester S, Dumais S T, Furnas G W, et al. Indexing by latent semantic analysis. Journal of the American society for information science. 1990, 41(6): 391-407.
52. Rohde D L T, Gonnerman L M, Plaut D C. An improved model of semantic similarity based on lexical co-occurrence. Communications of the ACM. 2006, 8(627-633): 116.
53. Bengio Y, Ducharme R, Vincent P, et al. A neural probabilistic language model. The journal of machine learning research. 2003, 3: 1137-1155.
54. Tomás Mikolov, Martin Karafiát, Lukás Burget, et al. Recurrent neural network based language model. In: Conference of the International Speech Communication Association (INTERSPEECH). Makuhari, Chiba, Japan, 26-30 September 2010, ISCA 2010: 1045-1048.
55. Jelinek F, Mercer R L, Bahl L R, et al. Perplexity—a measure of the difficulty of speech recognition tasks. The Journal of the Acoustical Society of America. 1977, 62(S1): S63-S63.
56. Papineni K, Roukos S, Ward T, et al. Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th annual meeting of the Association for Computational Linguistics. Philadelphia, PA, USA, 6-12 July 2002, ACL 2002: 311-318.
57. Levenshtein V I. Binary codes capable of correcting deletions, insertions, and reversals. Soviet physics doklady. 1966, 10(8): 707-710.
58. Brown P F, Cocke J, Della Pietra S A, et al. A statistical approach to machine translation. Computational linguistics. 1990, 16(2): 79-85.
59. Heafield K. KenLM: Faster and smaller language model queries. In: Proceedings of the sixth workshop on statistical machine translation. Edinburgh, Scotland, UK, 30-31 July 2011, Association for Computational Linguistics 2011: 187-197.
60. Koehn P. Europarl. A parallel corpus for statistical machine translation. MT summit. 2005, 5: 79-86.
61. Fuli Luo, Peng Li, Jie Zhou, et al. A Dual Reinforcement Learning Framework for Unsupervised Text Style Transfer. In: International Joint Conference on Artificial Intelligence (IJCAI). Macao, China, 10-16 August 2019, IJCAI 2019: 5116-5122.

# 附录1 攻读硕士学位期间参与的科研项目

[1] 海量多源异构数据的使用授权与鉴权体系研究（国家自然科学基金联合重点项目，U1836204）（2019-2022）.

[2] 大规模网络\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*技术研究（国家重点研发计划课题，2016QY01W0202）（2016-2019）.

# 附录2 攻读硕士学位期间发表的论文

[1] **Lulu Xiao**, Xiaoye Qu, Ruixuan Li, Jun Wang, Pan Zhou, Yuhua Li. Fine-Grained Text Sentiment Transfer via Dependency Parsing. In: the 24th European Conference Artificial Intelligence. 2020: 2228-2235.（CCF推荐B类会议，署名华中科技大学为第一单位）.

1. https://stanfordnlp.github.io/CoreNLP/ [↑](#footnote-ref-1)
2. https://pypi.org/project/spacy/ [↑](#footnote-ref-2)
3. https://github.com/FudanNLP/fnlp/wiki [↑](#footnote-ref-3)