



本科毕业设计(论文)

(辅 修)

学院(部) 计算机科学与技术学院

题 目 面向复杂分类体系的
多标签情绪分类算法设计与实现

年 级 2019 专业 生物信息学

班 级 生物信息学 学号 1930401096

姓 名 唐柳健

指导老师 朱苏阳 职称 讲师

论文提交日期 2023/5/9

苏州大学 本科毕业设计（论文）独创性声明

本人郑重声明：所提交的本科毕业设计（论文）是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本设计（论文）不含其他个人或集体已经发表或撰写过的研究成果。对本文的研究作出重要贡献的个人和集体，均已在文中以明确方式标明。本人承担本声明的法律责任。

作者签名：

唐柳健

日期：

2023/5/9

苏州大学 本科毕业设计（论文）使用授权声明

本人完全了解苏州大学关于收集、保存和使用本科毕业设计（论文）的规定，即：本科毕业设计（论文）的著作权以及文中研究成果的知识产权归属苏州大学。苏州大学有权向国家有关部门或第三方机构送交毕业设计（论文）的复印件和电子文档，允许毕业设计（论文）被查阅和借阅，可以采用影印、缩印或其他复制手段保存和汇编毕业设计（论文），可以将毕业设计（论文）的全部或部分内容编入有关数据库进行检索。

涉密设计（论文）☐

本设计（论文）属 _____ 在 _____ 年 _____ 月解密后适用本规定。

非涉密设计（论文）☒

论文作者签名： _____ 日期： 2023/5/9

导师签名： _____ 日期： 2023/5/9

面向复杂分类体系的多标签情绪分类算法设计与实现

摘要：

随着互联网技术的发展，社交媒体平台的流行，使得越多活跃网民涌入，在这些平台上发表各式各样的言论。这些评论中不乏包含着大量的价值观，情绪等有价值的信息待挖掘，分析这些信息可以应用于抑郁情感识别，民意检测等有用的任务。一段评论中，经常含有多种情绪，且经常含有表情符号、缩略语和俚语等特殊语言，使得社交媒体上包含大量情绪的多标签情绪分类仍是一个具有挑战性的任务。

本文旨在面对情绪类别较多的复杂体系下，利用情绪类别的相关性从不同情绪的层次关系出发，建立多层次模型获得比单层次更好的分类效果。本文多标签分类的策略采用 BR 方法，将一个多标签分类任务建模为若干个单标签分类任务，使用基于深度学习的模型对每个层次的情绪构建了二元分类器。在分类器选择方面，进行实验分析对比了 CNN，LSTM，BiLSTM，self-Attention，Attention+BiLSTM 的 5 种常用的深度学习模型在同样情绪下的分类效果，选择效果最佳的模型作为基模型构建多层次模型。此外，针对多层次模型预测出的空标签问题，为了最大程度发挥多层次的优势，使用整数线性规划（ILP）对多层次模型进行了改进与优化。最终在 GoEmotions 数据集上，划分了三个层次的情绪体系，从粗粒度到细粒度，依次为三元情绪体系，ekman 情绪体系，27 种细粒度情绪体系，共 36 种情绪，每种情绪都训练了上述 5 种基模型，共 180 个模型。

通过对比实验，5 种基模型在 36 种情绪的分类效果相差不大，LSTM 的效果略好，使用 LSTM 作为基模型构建了单层次模型，多层次模型，仅在第三层使用 ILP 的改进的多层次模型，在 5427 个样本的测试集上进行了比较。多层次模型比单层次模型在准确率和 F1 值上都提高了约 0.05。改进的多层次模型效果也比多层次模型得到了提高，仅在第三层使用 ILP 的模型相对原始多层次模型在准确率上提高了约 0.02，F1 上提高了约 0.03，而从第二层开始使用 ILP 的模型相对原始多层次模型在准确率和 F1 都上提高了约 0.05，与单层次相比提高了 0.1。最后通过，多标签数预测的对比和实际查看分析，进一步验证了多层次模型相比单层次模型的有效性。

关键词：多标签情绪分类；深度学习；多层次；整数线性规划

Design and implementation of multi-label emotion classification algorithm for complex classification system

Abstract:

With the development of Internet technology and the popularity of social media platforms, more and more active netizens flood into these platforms to express various opinions. These comments contain a large number of values, emotions and other valuable information to be mined, and the analysis of these information can be applied to depression emotion recognition, public opinion detection and other useful tasks. A comment often contains multiple emotions, and often contains special language such as emoticons, abbreviations and slang, which makes multi-label emotion classification with a large number of emotions on social media still a challenging task.

The purpose of this paper is to use the correlation of emotion categories from the hierarchical relationship of different emotions to establish a multi-level model to obtain better classification results than a single level under a complex system with many emotion categories. The strategy of multi-label classification in this paper adopts BR method, which models a multi-label classification task as a number of single-label classification tasks, and uses a deep learning-based model to construct a binary classifier for each level of emotion. In terms of classifier selection, the classification effect of five commonly used deep learning models including CNN, LSTM, BiLSTM, self-Attention, and Attention+BiLSTM under the same emotion is analyzed and compared, and the model with the best effect is selected as the base model to construct a multi-level model. In addition, aiming at the empty label problem predicted by the multi-level model, in order to maximize the advantages of multi-level, the multi-level model was improved and optimized by using Integer Linear Programming (ILP). Finally, on the GoEmotions dataset, we divided three levels of emotion systems, from coarse to fine granularity, which are ternary emotion system, ekman emotion system, and 27 fine-grained emotion systems, totaling 36 emotions. For each emotion, we trained the above five base models, totaling 180 models.

Through comparative experiments, the classification effect of the five base models on 36 emotions is similar, and the effect of LSTM is slightly better. LSTM is used as the base model to construct a single-level model, a multi-level model, and an improved multi-level

model using ILP only in the third layer, on the test set of 5427 samples. The multi-level model improves the accuracy and F1 value by about 0.05 compared with the single-level model. The effect of the improved multi-level model is also improved compared with the multi-level model. The model using ILP only in the third layer improves the accuracy by about 0.02 and F1 by about 0.03 compared with the original multi-level model, and the model using ILP from the second layer improves the accuracy and F1 by about 0.05 compared with the original multi-level model. Compared with the single layer, the improvement is 0.1. Finally, through the comparison of multi-label prediction and actual viewing analysis, the effectiveness of the multi-level model compared with the single-level model is further verified.

Keywords: Multi-label emotion classification; Deep learning; Multi-level; Integer linear programming

目 录

前 言	1
第 1 章 绪 论	3
1.1 研究背景及意义	3
1.2 研究现状	4
1.3 论文主要工作	5
1.4 论文结构安排	6
第 2 章 相关工作	8
2.1 多标签情绪分类任务	8
2.1.1 多标签情绪分类介绍	8
2.1.2 GoEmotions 数据集	8
2.1.3 评价指标	9
2.2 情绪体系及特点	10
2.3 情绪分类模型	10
2.3.1 CNN 模型	11
2.3.2 LSTM 模型及 BiLSTM 模型	11
2.3.3 self-Attention 模型及 BiLSTM+self-Attention 模型	12
2.4 本章小结	14
第 3 章 多层次模型的建立及改进	15
3.1 多层次模型构建	15
3.2 基模型的构建	16
3.2.1 词嵌入	17
3.2.2 五种情绪分类模型的具体网络框架	17
3.3 基于整数线性规划的改进	18
3.4 本章小结	21

第 4 章 多层次模型实现及实验结果分析	22
4.1 多层次模型实现	22
4.1.1 实验环境	22
4.1.2 实验数据及文本预处理	22
4.1.3 基模型的选择与对比	24
4.1.4 实现过程	25
4.2 多层次模型改进实现	26
4.3 实验结果与分析	27
4.4 本章小结	29
第 5 章 总结与展望	30
5.1 本文总结	30
5.2 后期工作与展望	30
参考文献	32
致 谢	34

前言

随着计算机算力和人工智能的发展，自然语言处理（Natural Language Processing, NLP）领域也因为纷纷出炉的深度学习技术而得到革新，越来越多的 NLP 任务研究受到了关注。同时，互联网技术的进步，各大社交媒体的蓬勃发展，每日都有拥有着巨大的活跃用户量，越来越多的用户在社交平台上发表自己的评论。这些评论中不乏包含着大量的价值观，情绪等有价值的信息待挖掘，分析这些信息可以应用于抑郁情感识别，民意检测等有用的任务。

通常，人们发表的言论文本，经常含有多种情绪，这些情绪中存在关联性，同时由于社交媒体上的短文本经常含有表情符号、缩略语和俚语等特殊语言，导致与传统的文本结构不同，并且文本中情感表达的语义模糊和情感类别的边界模糊，较单情绪的识别分类难，使得社交媒体上包含大量情绪的多标签情绪分类仍是一个具有挑战性的任务。

多标签情绪分类任务，到目前为止已有多种模型提出用于解决。基本的方法是使用深度学习方法，自动提取文本的特征与语义，连接各种网络层进行分类，但是都是多为单层次的方法，即认为各情绪类别间是独立的，在同一个情绪体系下进行分类。单层次方法在更细粒度，情绪类别较多的情况下往往出现分类准确率不高的情况。

本文旨在面对情绪类别较多的复杂体系下，利用情绪类别的相关性从不同情绪的层次关系出发，建立多层次模型获得比单层次更好的分类效果。本文多标签分类的策略采用 BR 方法，将一个多标签分类任务建模为若干个单标签分类任务，使用基于深度学习的模型对每个层次的情绪构建了二元分类器。在分类器选择方面，进行实验分析对比了 CNN, LSTM, BiLSTM, self-Attention, Attention+BiLSTM 的 5 种常用的深度学习模型在同样情绪下的分类效果，选择效果最佳的模型作为基模型构建多层次模型。此外，为了最大程度发挥多层次的优势，使用整数线性规划（integral linear programming, ILP）对多层次模型进行了改进与优化。最后，设置了一系列对比实验，在 GoEmotions 数据集上，通过比较评价指标，验证了本文方法提出的有效性。

本文的主要工作及贡献如下：

- （1）分析实验对比了常用于情感分类的 5 种深度学习模型，通过比较在相同情

绪上的准确率，选定了用于构建多层次模型的基模型。

（2）根据多层次模型的原理，对数据集进行了情绪体系分层，构建实现了多层次模型和单层次模型。

（3）根据多层次模型的局限性对其进行了改进，采用整数线性规划改善了多层次模型的空标签预测情况，并提高了准确率与 F1 值。

（4）在 GoEmotions 数据集上进行了评估。在效果上，多层次模型比单层次模型有更好的准确率与 F1 值，且多层次模型有着比单层次模型更快的预测速率。此外，优化后的多层次模型在准确率和 F1 值上都有更高的提升。

第1章 绪论

本章首先陈述了多标签情绪分类的研究背景和意义，其次简单介绍了该任务的研究现状，概述了本文完成的主要工作和贡献。最后，本章还展示了本文的整体组织结构。

1.1 研究背景及意义

随着计算机技术和互联网技术的提高，智能手机以及个人电脑的普及，社交媒体平台得到了前所未有的蓬勃发展。据 Statista 最新数据统计，截止 2023 年 1 月，全球使用互联网的用户已达到 51.6 亿，其中使用社交媒体的用户就达 47.6 亿，占世界人口的 59.4%。Instagram, Twitter, Reddit 等热门社交媒体的出现，贡献了巨大的用户量，这些用户来自世界各地，不同的年龄段，从事着各式各样的行业。社交媒体展现出强大的连接力，将全球的人们聚在平台的层出不穷的话题下，分享他们的有趣的想法和评论。随着更多人的涌入尤其是越来越年轻的活跃用户，使得更多简短的评论用来表达他们对某一话题的看法，这些评论包含着他们的价值观以及情绪等有挖掘意义的信息。

鉴于语言的趣味性，表达的多样性，富含的意义，挖掘社交网络上海量的文本变得极具价值，特别是对社交媒体上的短文本的多种情绪的分析，使得其成为自然语言处理（Natural Language Processing, NLP）的重要研究方向。情绪识别分类在生活中已有很多实用价值的应用，如对 Twitter 评论自动检测抑郁情感来预防抑郁症^[1]，针对网络上的舆论政治倾向的民意检测^[2]，在演讲中的由多情绪感知获得演讲者关注部分^[3]等。

情绪是一种内部的主观体验，即根据自身的需要或倾向为中介的心理现象。很多心理学家认为人的情绪由几个基本的情绪构成。据不同心理学家的，情绪可以分为二元情绪（包括中性情绪），普鲁契克情绪轮^[4]，Ekaman 情绪^[5]等离散型情绪。也有学者认为情绪是不可分割的，像光的色谱一样是连续变化的，提出了将情绪看作三维空间中的点的 VAD 模型^[6]等的连续型情绪。通常一个句子中包含多种情绪，人们会

组合各种情绪表达更为复杂的情感。如表 1.1 所示，往往一个短的句子中就包含着多种情绪。再者，由于社交媒体上的短文本经常含有表情符号、缩略语和俚语等特殊语言，导致与传统的文本结构不同，并且文本中情感表达的语义模糊和情感类别的边界模糊，较单情绪的识别分类难，使短文本的多标签情绪分类仍是一个具有挑战性的任务。

表 1.1 GoEmotions 数据集中 Reddit 评论与对应情绪

Reddit 评论	标签
aw, thanks! I appreciate that!	admiration,gratitude
Holy shit was that the one guy that supported remain? I didn't know we had a celebrity on this	confusion,curiosity,disappointment,surprise
I'm so excited for this game!! ☺	excitement
I didn't know it was gonna be that simple	confusion
Forgive me [NAME] :D	
That guy is being intentionally obtuse. Down vote trolls are a waste of everyone's time.	annoyance,disapproval

综上，无论是应用还是解决问题的难度上，多标签情绪分类都是一项值得研究与挑战的问题。

1.2 研究现状

多标签情绪分类可以看作是多标签分类与情绪分类两个任务的交叉问题。多标签分类在传统上，采用问题转换和算法自适应来解决。问题转换方法将多标签分类转换为常见的分类任务。经典的问题转换方法包括二元关系法（BR）^[7]，分类器链（CC）^[8]和标签幂集（LP）^[9]等。BR 将多标签分类转为单标签分类，单独预测每个标签，缺点在于忽略了标签的关联性。CC 转化为链式的多个二分类问题，同时考虑了标签的共现性，但存在预测时的暴露误差的缺陷。LP 将每个可能的标签组合视为一个单独的标签，因此它可能会将标签数量呈幂级增长而不可用。算法自适应方法则扩展了现有的单标签分类算法以处理多标签分类。对于情绪分类，早期采用精心设计的人工特征集如词典，词袋，特定的语义规则等，用于多标签分类上。但是该做法往往需要大量的专家知识，费时费力，且质量高度依赖于注释的好坏。

随着深度学习的发展，由于深度神经网络能很好的学习和自动提取特征，使用深度学习技术的方法越来越受到关注。随着越来越多的实验证明，像循环神经网络（RNN）以及配有门控循环单元层（GRU）的长短期记忆网络（LSTM），双向长短期记忆网络（BiLSTM），以及使用注意力机制等的组合在学习短语级特征时非常有用^[10]，有更好的语言理解能力与语义表示，在情感分类中取得了非常好的精度。因此有很多基于或更多神经网络的变种被提出用于多标签情绪分类。

He 等人^[11]结合二元相关神经网络（BRNN）和阈值依赖神经网络（TDNN）的各自优势，提出了联合二元神经网络（JBNN）改进了 BR 方法。在 JBNN 中，文本表示被送入一组 logistic 函数而非 softmax 函数，多个二分类在一个神经网络框架中同步进行。基于注意力机制上，Kim 等人^[12]使用由一个自注意力模块和多个 cnn 组成，模仿人类分析句子的过程进行情绪分类，但由于使用的独立的 cnn 进行分类，未考虑到标签的相关性。Jabreel 等人^[13]还提出了一种新的转换机制且基于注意力机制的深度学习系统二值神经网络（BNet），该系统由数据驱动的、基于神经的端到端模型，不依赖于词性标记器和情感词典等额外的专家注释。在使用自注意力机制基础上，Baziotis 等人^[14]利用迁移学习，对他们的配备多层自注意力机制的 Bi-LSTM 架构进行了预训练，补偿了有限的训练数据，利用网络的编码层通过在正式分类任务上微调取得较好结果。

但是，上述方法都未考虑到情绪体系的层次性，本文针对之一问题，结合经典的深度学习训练器从多层次的角度进行多标签情绪的分类的研究。

1.3 论文主要工作

本文以多标签情绪分类为问题导向，旨在面对情绪类别较多的复杂体系下，利用情绪类别的相关性从不同情绪的层次关系出发，建立多层次模型获得比单层次更好的分类效果。本文多标签分类的策略采用 BR 方法，将一个多标签分类任务建模为若干个单标签分类任务，使用基于深度学习的模型对每个层次的情绪构建了二元分类器。在分类器选择方面，进行实验分析对比了 CNN，LSTM，BiLSTM，self-Attention，BiLSTM+self-Attention 的 5 种常用的深度学习模型在同样情绪下的分类效果，选择效果最佳的模型作为基模型构建多层次模型。此外，为了最大程度发挥多层次的优势，

使用整数线性规划（integral linear programming, ILP）对多层次模型进行了改进与优化。最后，设置了一系列对比实验，在 GoEmotions 数据集上，通过比较评价指标，验证了本文方法提出的有效性。

本文的主要工作及贡献如下：

（5）分析实验对比了常用于情感分类的 5 种深度学习模型，通过比较在相同情绪上的准确率，选定了用于构建多层次模型的基模型。

（6）根据多层次模型的原理，对数据集进行了情绪体系分层，构建实现了多层次模型和单层次模型。

（7）根据多层次模型的局限性对其进行了改进，采用整数线性规划改善了多层次模型的空标签预测情况，并提高了准确率与 F1 值。

（8）在 GoEmotions 数据集上进行了评估。在效果上，多层次模型比单层次模型有更好的准确率与 F1 值，且多层次模型有着比单层次模型更快的预测速率。此外，优化后的多层次模型在准确率和 F1 值上都有更高的提升。

1.4 论文结构安排

本文共分为五章，各章内容安排如下：

第一章：绪论。介绍了本文所述课题的研究背景和意义，简单地介绍了多标签情绪分类的研究现状和难点，以及本文完成的主要工作和贡献，最后介绍了本文的组织结构。

第二章：相关工作。阐述了多标签情绪分类的基本定义，本文采用的数据集以及评价指标，然后简单的介绍了经典的情绪体系，最后详细的介绍了 5 种常用的基于深度学习的情感分类模型并在第四章给出了它们的效果。

第三章：多层次模型的建立与改进。提出了基于多层次模型的多标签情绪分类方法。详细介绍了模型的整体框架和每个模块。最后提出了采用整数线性规划改进的原理和实现方法。

第四章：多层次模型实现及实验结果分析。首先给出了二元分类模型的通用的预处理方法，然后给出了 5 种情感分类模型的实现和搭建方法，测试并展示了它们训练

后的分类效果。然后使用最佳的模型构建多层次模型和单层次模型以及优化后的多层次模型，描述了它们在数据集上的测试效果，并分析了实验结果。

第五章：总结与展望。提出了一些关于该课题的未来工作，可补充内容以及展望。

第 2 章 相关工作

本章先介绍了多标签情绪分类任务的基本概念定义，介绍了本文采用的数据集，以及评价指标，然后简单阐述了现有的情绪体系，最后详细阐述了 5 种常用于情感分类的深度学习模型，该 5 种模型将用于后面的对比实验。

2.1 多标签情绪分类任务

2.1.1 多标签情绪分类介绍

多标签情绪分类可以定义为，在给定的数据集 $D = \{(x^{(i)}, y^{(i)}) | i = 1, \dots, n\}$ ，其中 n 为样本数， $x^{(i)} \in R^d$ 的样本空间。假设 $E = \{e_1, e_2, \dots, e_m\}$ 为数据集可能的情绪标签集，每个样本的 $y^{(i)}$ 可以被定义为 $y^{(i)} = \{y_1, y_2, \dots, y_k\}$ ，其中 $k \leq m$ 。最终要求模型能从文本提取出语义特征 $x^{(i)}$ 并在训练集上学习到 $f(x^{(i)}) \rightarrow y^{(i)}$ 的映射。

2.1.2 GoEmotions 数据集

本文采用 Google 研究员制作的 GoEmotions 数据集^[15]。GoEmotions 数据集与传统的数据集（大多仅包含 6 种基本情绪）不同，它更加细粒度，包含了 27 种类别的情绪与中性情绪，共 28 种情绪，适用于本文面向复杂分类的体系。GoEmotions 数据集采样于知名社交平台 Reddit 的各种不同话题板块，包含 58k 的评论，并且经过了精细的人工注释，为高质量的情感分类数据集。

GoEmotions 的文本样例可见表 1.1，每个评论都包含了不同种情绪，有些评论虽然短，但是表达了复杂的情绪，光靠简单的模型是很难区别的。研究人员在构建数据集时，考虑了心理学因素和数据的适用性，在 Ekman 情绪体系上进行了扩大。并且在事先没有规定情绪概念的情况下，27 种情绪由相关性构建聚类后得到了 12 种积极的，11 种消极的，4 种模糊的情绪聚类，表明了这些情绪间的关联性，如图 2.1 所示，可以通过聚类树发现除了聚类成了三大情感外，相同集群下情感间也存在不同的关联

性，表明了还可以继续再分层的可能性，这也是对本文的启发。除了在情绪类别上，研究人员为了达到最大程度覆盖 Reddit 上所表达的情绪，对评论选择上也增加了限制，在板块话题进行了平衡，确保热门板块不占比过大而能反映冷门板块中评论所带有的情绪。此外，也删除了一些有害的，不健康的，有偏见的言论，使得整个数据集具有广泛性。

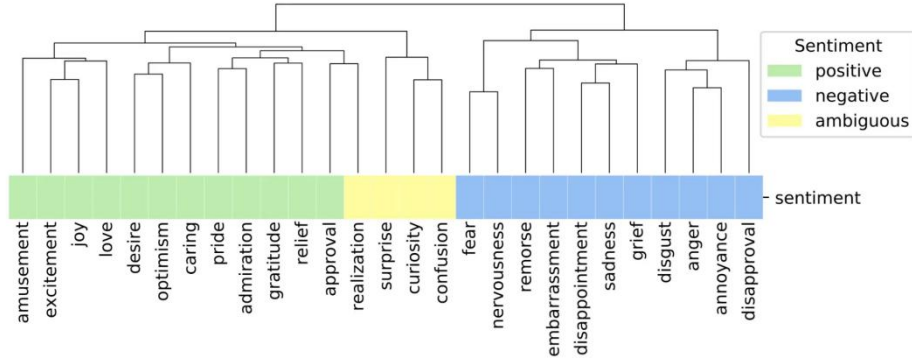


图 2.1 GoEmotions27 种情绪聚类图

2.1.3 评价指标

本文评价指标采用，多标签分类任务下的准确率与 F1 值。计算公式(2.1-2.2)如下：

$$Acc = \frac{1}{n} \sum_{i=1}^n \frac{|y_i \cap y'_i|}{|y_i \cup y'_i|} \quad (2.1)$$

$$F1 = \frac{2}{n} \sum_{i=1}^n \frac{|y_i \cap y'_i|}{|y_i| + |y'_i|} \quad (2.2)$$

其中 n 为被测试的样本总数， y_i 是样本的真实类别标签集合（如样本 i 的真实标签为 $[a, b, c]$ ，则该集合为 $\{a, b, c\}$ ）。 y'_i 是通过分类模型预测得到的标签集合（如样本 i 的真实标签为 $[a, g]$ ，则该集合为 $\{a, g\}$ ）。 $|\cdot|$ 用于返回一个集合的大小。准确率可以直观的反映分类的性能，而 F1 值为模型准确率与召回率（实际为正的样本中被预测为正样本的概率）的调和平均，当两个模型的准确率和召回率各自有优势时可以用 F1 来综合衡量。本文通过上述指标，通过对比所设计的多层次模型的性能与只使用单层次模型的性能以及改进的多层次模型，来验证方法的有效性。

2.2 情绪体系及特点

经典的情绪体系有，Ekman 情绪体系，Plutchik 情绪体系。Ekman 认为人们对于面部表情和情绪的匹配具有高度的一致性。因为大家往往将充满笑意的脸与“快乐”联系起来，将紧皱眉头、紧闭双眼的面孔与“愤怒”相匹配。Ekman 通过实验给不同文化背景的实验者展示人脸照片并且让其对应一个情绪的方法总结出了 6 个普遍的基本情绪，分别为快乐、愤怒、恐惧、悲伤、恶心、惊讶。Plutchik 则提出了 8 个基本情绪，分别为喜悦、信任、恐惧、惊讶、悲伤、期待、愤怒和厌恶。并且用著名的 Plutchik 情绪轮，如图 2.2，将这些情绪按照对立（如悲伤与快乐），组合（没有颜色的部分如将快乐和期待组合为乐观），强度（颜色越深，情绪越强烈）等方面展现了这些情绪的复杂的关系。

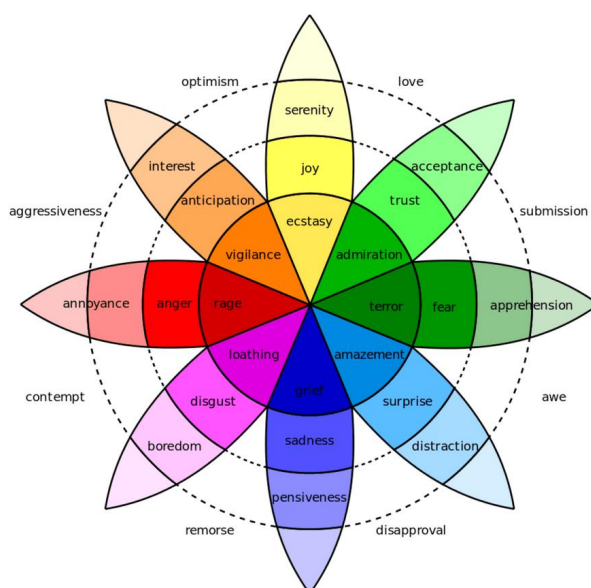


图 2.2 Plutchik 情绪轮

2.3 情绪分类模型

到目前为止，已有很多深度学习技术用于情绪分类上，本文主要采用这些典型的模型或这些模型的组合，意在提取出更好的语义特征或是编码出更好的思想向量用于下游的分类任务。

2.3.1 CNN 模型

CNN（Convolutional neural net）最大的特点为在卷积层采用滑动窗口（卷积核）对数据进行类似扫描的处理。每个滑动窗口，即卷积核，每个位置都有相应的权重，可以对该区域内进行卷积运算再由一个激活函数输出，从而能捕捉到该区域内数据点的空间关系（时序关系），提取到该区域的特征。CNN 最早是用于图像处理，简单来说，卷积核在图像上滑动后，可以得到模糊的，保留了主要特征的图像。由于 CNN 的特性，通常可用于时序数据，所以也可以压缩为一维应用在自然语言处理上，对文本进行滑动，提取语义。卷积核会随着网络的训练和反馈更新权重，因此卷积核如何提取特征也可通过学习完成。如图 2.3 所示，卷积核的宽度为 3，以一个步长对文本进行滑动提取语义特征。

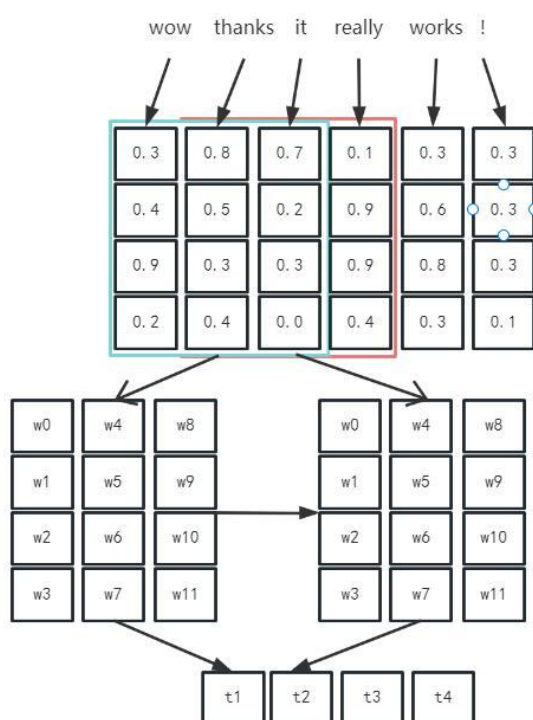


图 2.3 CNN 滑动窗口示意

2.3.2 LSTM 模型及 BiLSTM 模型

LSTM（Long short-term memory）网络为 RNN（Recurrent neural net）的改进版本。RNN 利用 t 时刻的输出作为 $t+1$ 时刻的输入而引入了循环回路将网络进行了“展开”，

使得网络具有记忆功能，可以记住句子中过往出现的信息。但是 RNN 有个致命的特点是随着句子序列长度的增加，根据反馈更新权重时会面临梯度爆炸或消失等问题。为此，LSTM 在 RNN 的基础上使用门控循环单元（gated recurrent unit, GRU），使得有遗忘部分不重要的记忆的功能。LSTM 在每一层额外输入类似于“记忆”的一种状态信息，并且该状态信息随着训练可以更新。LSTM 的每个单元可以看作由遗忘控件，候选控件，输出控件三个控件够成，如图所示。遗忘控件通过掩码来选择遗忘部分记忆状态，而候选门控件用于学习和记得新的记忆状态来更新，最后输出控件用于拼接上一时刻的输出和当前时刻的输入。因此 LSTM 利用有舍弃的记忆来避免了 RNN 的弊端。

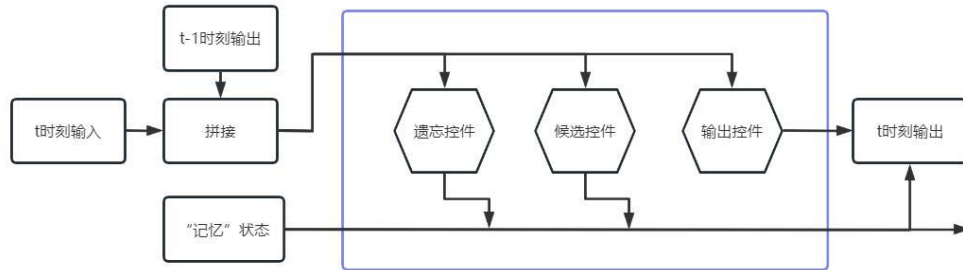


图 2.4 LSTM 结构示意图

BiLSTM 网络即为使用双向的 LSTM 的网络。由于句子中的词关系往往需要逆向的追溯才能理解句子的涵义，要结合正向和反向的词关系信息。BiLSTM 在正向 LSTM 的基础上将展开的网络的最后一个词对应的隐藏层作为第一时刻的输入反向传播到第一个词的隐藏层。通过该方法，以期望对文本句子能有更深刻，模仿人类解读的效果。

2.3.3 self-Attention 模型及 BiLSTM+self-Attention 模型

注意力机制一开始由模仿人类看图片时眼睛注视的焦点而忽略无关紧要的部分而被提出。在计算机视觉与自然语言处理上都有广泛的应用。注意力机制一般用于 seq2seq 模型。在普通的 seq2seq 模型中，输入为 (x_1, x_2, \dots, x_n) 及输出为 (y_1, y_2, \dots, y_n) 的 seq2seq 模型经过编码层后出来的语义编码为 $C = f(x_1, x_2, \dots, x_n)$ ，将其传入到解码层后第 i 个输出为 $y_i = g(C, y_1, y_2, \dots, y_{i-1})$ ，可以看到不管 i 为多少，语义编码都是相同的，

但是显然有些句子中的各别单词对输出影响程度是不同的，要引入注意力，针对不同的单词生成出的语义编码是不同的。因此注意力机制的原理如下，对于每个 i 都有对应的语义编码，见公式(2.3)：

$$y_i = g(C_i, y_1, y_2, \dots, y_{i-1}) \quad (2.3)$$

而整个句子的语义编码 C_i 可由输入的句子中每个单词的语义编码乘上相应的权重求和表示，见公式(2.4)：

$$C_i = \sum_{j=1}^m a_{i,j} h_j \quad (2.4)$$

其中， h_j 为第 j 个单词的隐藏状态（即语义表示），而权重 $a_{i,j}$ 可由中间值 $e_{i,j}$ 进行 softmax 计算得到，见公式(2.5)：

$$a_{i,j} = \frac{\exp(e_{i,j})}{\sum_{s=1}^n \exp(e_{i,s})} \quad (2.5)$$

中间值 $e_{i,j}$ 由输入与每个词的隐藏状态值交互计算得到，见公式(2.6)：

$$e_{i,j} = h_j \tanh(WX_i) \quad (2.6)$$

其中 W 为整个模型需要学习的神经元权重矩阵， \tanh 为激活函数。

整个过程可以概括为输入句子与目标句子的对齐过程，即为字典一样的映射查询过程，可看作为计算查询向量(Query)与键(Key)的相似度作为权重与候选值(Value)加权运算求和得到注意力值。

自注意力 (self-Attention) 机制^[16]为上述注意力机制上输入句子与输出句子为同一个句子特例情况。因此使用 self-Attention 可以捕获一个句子中的句法特征，短语结构，也能得到一定距离的句子中词义的“记忆”关系，可以记住远距离相互依赖的特征，进而学习到句子的语义，可用于情绪分类任务中。

BiLSTM+Attention 模型为即本文上述的 self-Attention 模型与 BiLSTM 模型相结合的模型，以期望通过 BiLSTM 层对句子进行初步的语义特征编码后传给自注意力机制层进行进一步的关注重要的部分的语义编码，以期望获得更好的语义表示，用于本文的情绪分类任务。

2.4 本章小结

本章简单介绍了多标签情绪分类任务的定义以及本文使用的 GoEmotions 数据集，和本文使用的评价指标，接着概述了经典的情绪体系以及各自对应的特点，最后介绍了本文使用的 5 种情绪分类基模型的原理和本文使用的目的。

第3章 多层次模型的建立及改进

本章首先介绍了多层次模型的整体框架以及多层次模型使用的基模型的框架，最后针对多层次模型的缺陷提出了采用整数线性规划改进的原理和实现方法。

3.1 多层次模型构建

本文使用的 GoEmtions 可以从粗粒度到细粒度分为三个层次，第一层为 positive, negative, ambiguous 构成的三元情绪层，第二层为 Ekman 情绪体系层，第三层为 GoEmtions 的 27 种细粒度情绪层。因此，可以从这三个层次的体系出发，先进行粗粒度的分类，再分别进行下一层的细粒度分类。例如，给定一个文本 t ，先对其进行最高层次的 [A, B, C] 三分类。如分为 B 类，则对再 B 类下的 [c, d, e, f, g] 五个小类进行判别，得到 e、g 两类，以此类推，具体如下伪代码算法 1。本文以上述三层建立多层次模型，以 27+1 种情绪作为单层次模型。多层次模型具有比单层次明显的优势，多层次模型中的子分类器将不会考虑不在该大类里的情绪，因此在预测速度上比单层次快，且利用了情绪类别的关联，较少会出现预测出具有对立情绪的标签。

算法 1 多层次模型伪代码

```
1: for sample in Dataset
2:   for first_level_emo in 三元情绪
3:     f_pro = 预测 sample 属于 first_level_emo 的概率
4:     if f_pro > 0.5
5:       for second_level_emo in 属于对应 first_level_emo 类下的 Ekman 情绪
6:         s_pro = 预测 sample 属于对应 second_level_emo 的概率
7:         if s_pro > 0.5
8:           for third_level_emo in 属于对应 second_level_emo 类下的 27 种 GoEmotions 情绪
9:             t_pro = 预测 sample 属于 third_level_emo 的概率
10:            if t_pro > 0.5
11:              third_level_emo 添加到 sample 的预测标签集中
```

12: else if 都不属于三元情绪之一

13: neutral 情绪添加到 sample 的预测标签集中

多层次模型中的二元分类器即基模型采用在第二章所介绍的 5 种情绪分类模型中效果最好的一种，具体框架如图 3.1 所示，每个情绪类别由对应的基模型进行分类。单层次模型为该模型的最底层一层聚合在一起分别对 27+1（Neural）种情绪进行二元分类。

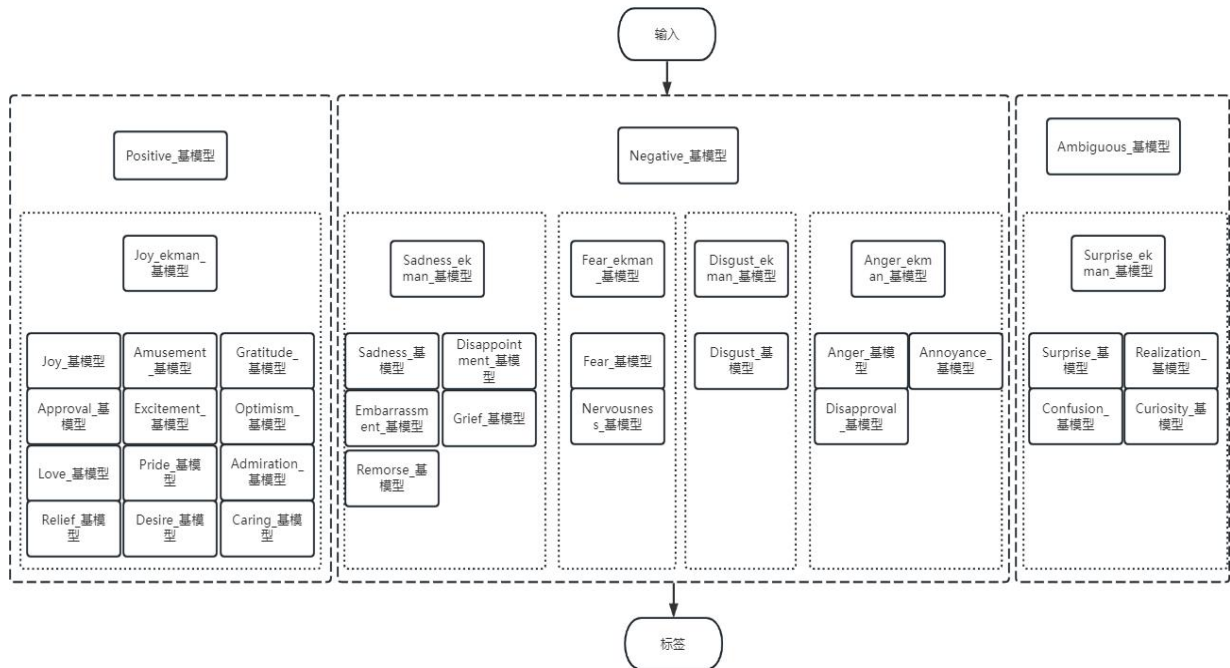


图 3.1 多层次模型框架

3.2 基模型的构建

本文采用第二章所介绍的五种常用基于深度学习的情绪分类模型，选择分类效果最好的一种作为多层次模型的基模型，基模型由预处理及词嵌入部分和各自的语义编码和分类结合成的两大部分够成。主要将句子进行分词和词嵌入，填充后输入到模型中进行训练学习到如何编码出能够有效代表句子的语义，之后进行预测分类。下面主要介绍，词嵌入部分和上述 5 种情绪分类模型的具体网络层结构。

3.2.1 词嵌入

由于直接输入单词是不能进行处理的，需要将单词映射为一个数值向量。但同时又希望映射出的向量能表示单词的含义，即语义相近的单词在向量空间上临近，可以通过计算夹角余弦值而得到它们的相似性。另外，若采用由词袋够成的多热或独热编码向量会造成数据稀疏的问题。面对上述问题，词嵌入向量被提出。词嵌入向量可以由两种方式来计算，skip-gram 方法和 CBOW(continuous bag-of-words)方法。Skip-gram 方法使用目标词来预测输出词来更新参数矩阵作为嵌入向量，而 CBOW 则使用临近词来预测目标词。现在主流的词嵌入模型有 Word2vec^[17]，Glove^[18]，fastText^[19]等。使用词嵌入模型可以将原本使用词袋的高维度向量压缩成较低维度的向量来表示单词，因此非常方便。

本文使用 Word2vec 对单词进行嵌入，对于未出现的词采用随机初始化的方法进行赋值。

3.2.2 五种情绪分类模型的具体网络框架

五种情绪分类模型的框架如图

CNN 模型框架具体使用一维的卷积层，采用 250 个宽度为 3 的卷积核，扫描步长为 1，使用“valid”方式进行填充，使用 relu 进行激活。接着连接一个一维的最大池化层进行粗略降维以及获得位置不变性，紧接着连接一个神经元个数为 250 个的全连接层后用 dropout 层以 20%比例进行丢弃来防止过拟合，再使用 relu 激活函数进行非线性化，最后连接一个神经元个数为 1 的 dense 层使用 sigmoid 函数获得预测的概率值。

LSTM 模型框架具体第一层使用 50 个神经元的 LSTM 层，接着以 20%比例而进行丢弃的 dropout 层，然后使用 Flatten 层将上一层输出全部压缩为一维用于下面采用一个神经元用 sigmoid 处理的输出层。

BiLSTM 模型框架为在 LSTM 模型框架上第一层改为双向 LSTM 层，其他保持一致。

self-Attention 模型框架为根据第三章所提的自注意力机制原理实现的注意力层

接着一个 Flatten 层压平输出到输出层。

BiLSTM+self-Attention 模型为第一层使用 50 个神经元的 BiLSTM 层，接着一个 50 个神经元采用 relu 激活的全连接层，并且接着一个以 20% 丢弃的 dropout 层。然后，接入自注意力机制层，接着一个 50 个神经元采用 relu 的全连接层，使用一个神经元的 dense 层用 sigmoid 激活输出预测概率值。

以上模型都有相同的输入维度的输入层，都采用 rmsprop 为优化器，二元交叉熵来计算 loss，训练时度量之指标使用准确率。

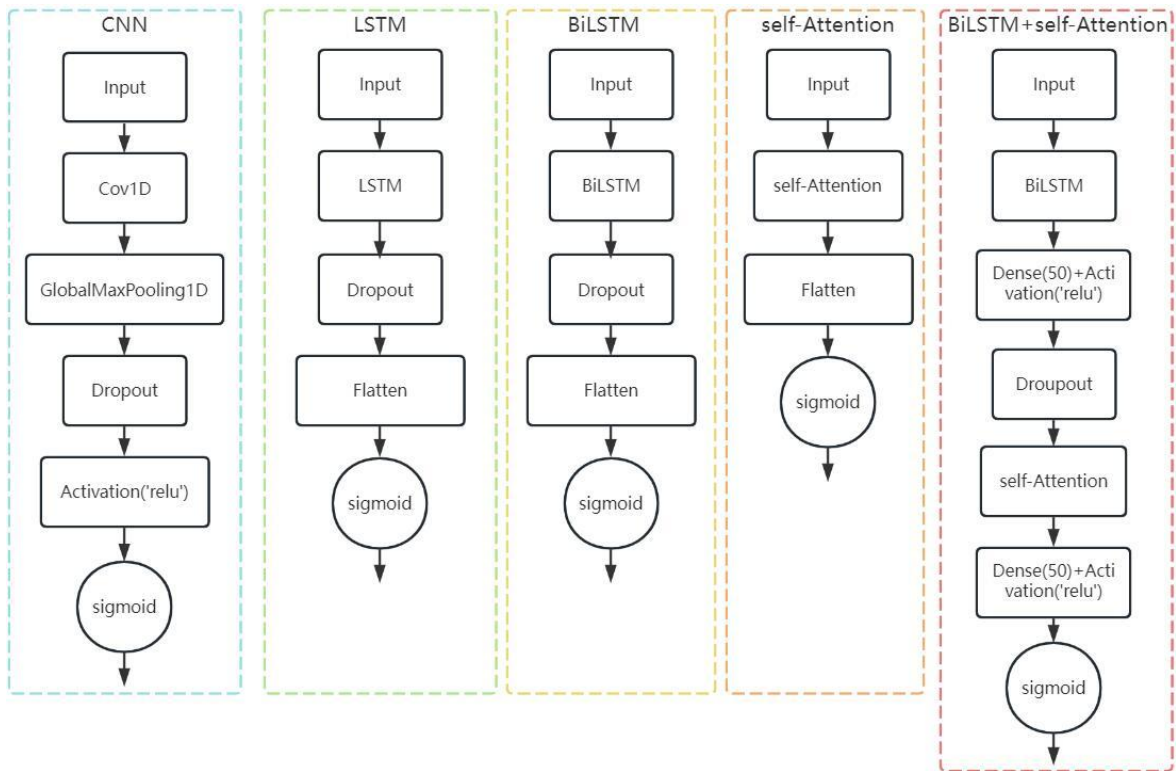


图 3.2 五种情绪分类模型框架

3.3 基于整数线性规划的改进

多层次模型在进行预测过程中，可能会出现最终预测出空标签的情况，这是由于在第二层中 6 个基本的 ekman 情绪预测出的概率值都低于 0.5，如图 3.3 所示虽然在第一层中预测属于 positive 类别，但是进入到属于 poitive 类别的第二层中没有预测出

大于 0.5 的 ekman 情绪。或者第二层中有预测值大于 0.5 的 ekman 情绪然而进入该情绪下的第三层后所有预测值都低于 0.5 造成没有标签预测出的情况，如图 3.4 所示，前两层都预测出了类别，但是在相应类别的第三层预测中没有大于 0.5 的预测值。

```
-----i-----: 12
Positive_LSTM: predicted-probability:0.8374795913696289 predicted-label:1
Joy_ekman_LSTM: predicted-probability:0.4755668342113495 predicted-label:0
Negative_LSTM: predicted-probability:0.17665740847587585 predicted-label:0
Ambiguous_LSTM: predicted-probability:0.08055838942527771 predicted-label:0
-----over-----
```

图 3.3 空标签情况 1

```
-----i-----: 2
Positive_LSTM: predicted-probability:0.92753005027771 predicted-label:1
Joy_ekman_LSTM: predicted-probability:0.9136620759963989 predicted-label:1
Joy_LSTM: predicted-probability:0.0884692370891571 predicted-label:0
Amusement_LSTM: predicted-probability:0.019895583391189575 predicted-label:0
Approval_LSTM: predicted-probability:0.07140126824378967 predicted-label:0
Excitement_LSTM: predicted-probability:0.22996234893798828 predicted-label:0
Gratitude_LSTM: predicted-probability:0.02351740002632141 predicted-label:0
Love_LSTM: predicted-probability:0.007670044898986816 predicted-label:0
Optimism_LSTM: predicted-probability:0.39467552304267883 predicted-label:0
Relief_LSTM: predicted-probability:0.00037735700607299805 predicted-label:0
Pride_LSTM: predicted-probability:0.0001621246337890625 predicted-label:0
Admiration_LSTM: predicted-probability:0.3239821791648865 predicted-label:0
Desire_LSTM: predicted-probability:0.01828068494796753 predicted-label:0
Caring_LSTM: predicted-probability:0.01294735074043274 predicted-label:0
Negative_LSTM: predicted-probability:0.0025353431701660156 predicted-label:0
Ambiguous_LSTM: predicted-probability:0.04477715492248535 predicted-label:0
-----over-----
```

图 3.4 空标签情况 2

针对上述问题，本文采用整数线性规划（Integer Linear Programming, ILP）进行改进。通过测试发现，大部分空标签的样本的真实标签对应的第三层情绪的预测值比同类下非真实标签的情绪的预测值要高很多且大部分游离在接近 0.5 附近，因此想通过 ILP 用于情绪预测改进的方法^[20]，对于空标签的情况能够输出这些更加接近 0.5 附近的情绪标签来改善多层次模型。具体如下：

$$c_i = -\log(p(e'_i = e_i)) + \log(1 - p(e'_i = e_i)) \quad (3.1)$$

$$C = (c_1, c_2, \dots, c_m)^T \quad (3.2)$$

$$L = (y_1, y_2, \dots, y_m)^T \quad (3.3)$$

$$\begin{aligned} \min z &= |L \times C| \\ &= \sum_{i=1}^m |c_i y_i| \end{aligned} \quad (3.4)$$

$$\text{s.t.} \quad y_i \in \{0, 1\} \quad (3.5)$$

$$1 \leq \sum_{i=1}^m y_i \leq 3 \quad (3.6)$$

其中， c_i 为当前样本预测出的情绪标签等于真实标签的代价， $p(e'_i = e_i)$ 代表当前预测属于第 i 情绪的概率值，即模型对该情绪的预测输出概率， m 为当前类下的情绪数量。 y_i 为预测出的标签，即 0 或 1 的二值，该变量为 ILP 中进行规划后输出的值。 z 为目标函数，约束条件为公式(3.5-3.6)所示。

该代价函数(3.1)可视化如图 3.5 下，在 0.5 处取到 0 值，结合到目标函数中可以理解为最优化的过程是尽量取到靠近 0.5 附近的点，选取的点即为输出的预测标签。通过该方法可以将空标签的问题进行优化，能够将焦点放在靠近预测概率值在 0.5 附近的标签，同时可以实现多标签的输出。如图 3.6 所示，当前类下为三个情绪，预测出的概率值为 P ， C 对应了计算出的对应代价，经过 ILP，最后选出第一个和最后一个情绪，都是靠近概率值为 0.5 附近的情绪，从而也能实现多标签的输出。

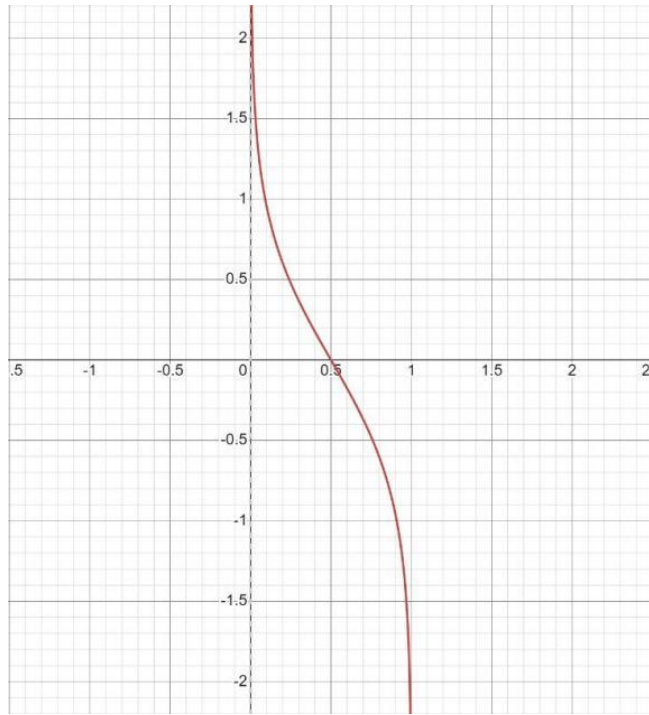


图 3.5 代价函数可视化

```
P = [0.6, 0.08, 0.42]
C = [-0.584962500721156, 3.523561956057013, 0.46566357234881217]
优化结果: 0.11929893
参数取值: [1, 0, 1]
```

图 3.6 ILP 例子

本文具体为输入该类下的预测概率值的列表，将 0.5 阈值修改为 0.65，对在第三层使用 ILP 和从第二层开始就使用 ILP 进行了对比，结果在第四章所示，引入 ILP 后比原先的多层次模型有了较好的提高效果。

3.4 本章小结

本章首先介绍了本文所提出的多层次模型的框架和具体构建，接着详细介绍了基模型的框架，以及 5 种情绪分类模型的具体网络结构，最后阐述了本文提出的使用 ILP 进行多层次改进的原理和方法。

第 4 章 多层次模型实现及实验结果分析

本章基于第 3 章的多层次模型原理和使用 ILP 的改进，对该方法进行了实现。挑选了最佳的基模型构建了多层次模型和单层次模型，最后通过测试集对比了多层次模型，改进后的多层次模型及单层次模型，以验证了本文提出方法的效果。

4.1 多层次模型实现

4.1.1 实验环境

- (1) 操作系统: Windows10
- (2) 处理器: Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz 1.80 GHz
- (3) RAM: 8.00 GB
- (4) 开发语言: Python3.9.0
- (5) 开发工具: PyCharm 2021.1 (Community Edition)
- (6) python 依赖包: numpy, keras, nltk, gensim, pandas, tensorflow, pulp

4.1.2 实验数据及文本预处理

本文模型在 GoEmotions 数据上进行训练和测试。GoEmotions 包含了 27+1 种情绪，详细情况已在 2.1.2 节给出。数据集按照训练集和开发集，测试集分成了三个 tsv 文件，数据集的具体划分情况如表 4.1。如图 4.1 所示，文件中的每行以制表符可以分割为三个部分，第一部分为文本本身，第二部分为类别序号（下标从 0 开始，类别顺序依次为 admiration、amusement、anger、annoyance、approval、caring、confusion、curiosity、desire、disappointment、disapproval、disgust、embarrassment、excitement、fear、gratitude、grief、joy、love、nervousness、optimism、pride、realization、relief、remorse、sadness、surprise、neutral），第三部分为文本来源地址（不考虑）。


```
My favourite food is anything I didn't have to cook myself. 27 eebbqej
Now if he does off himself, everyone will think hes having a laugh screwing with people instead of actually dead 27 ed00q6i
WHY THE FUCK IS BAYLESS ISOING 2 eezlygj
To make her feel threatened 14 ed7ypvh
Dirty Southern Wankers 3 ed0bdzj
OmG pEyToN iSn'T gOoD eNoUgH to hElP uS iN tHe PlAyOfFs! Dumbass Broncos fans circa December 2015. 26 edvnz26
Yes I heard abt the f bombs! That has to be why. Thanks for your reply:) until then hubby and I will anxiously wait @ 15 ee3b6wu
We need more boards and to create a bit more space for [NAME]. Then we'll be good. 8,20 ef4qmod
Damn youtube and outrage drama is super lucrative for reddit 0 ed8wbdn
It might be linked to the trust factor of your friend. 27 eczgvlo
Demographics? I don't know anybody under 35 who has cable tv. 6 eel6g5h
Aww... she'll probably come around eventually, I'm sure she was just jealous of [NAME]... I mean, what woman wouldn't be! lol 1,4
Hella awesome. Im from Toronto as well. Can call and visit in person if needed. 27 af02m1c
```

图 4.1 GoEmotions 原始数据示例

表 4.1 数据集划分

	多标签数	样本数
训练集	7102	43410
开发集	878	5426
测试集	837	5427

接着，需要对文本进行预处理，使用 NLTK 包对文本进行分词。采用 TreebankWordTokenizer 分词器，根据其内置的分词规则，将相邻单词分开，一些结束符号如“.:?! ”也会当成一个词条（token）分割。更重要的是能将缩略语进行拆分，如“don’t”拆开为“do”与“n’t”，若不进行拆分，不容易学到该词强调的“not”的否定语义，且可能会丢失“do”的含义。而 NLTK 库还提供了更适用于社交网络用语的 casual_tokenize 分词器，可以处理非常规的网络用语以及表情符号等，包括一些“Amazzzzziiiiinnngggg”之类的表达，可还原为正常用法。但经过实验对比发现，casual_tokenize 用于 GoEmotions 数据集上的文本时分词并不稳定，而且往往没有将缩略语进行拆分且 TreebankWordTokenizer 也能很好的将表情符号单独分割出，效果并没有 TreebankWordTokenizer 好。此外，GoEmotions 数据集已经将用户名，人名，宗教词 mask 为了[NAME]以及[RELIGION]的统一 token，这样可以减少因为这些特殊词带来的干扰，不过由于经过分词的缘故，这些 token 也被进行了拆分，本文重新将它们进行了拼接。最后将所有词进行了小写化，使得同样的词不会因为大小不一样而被当成不同的词。

之后使用 Word2Vec 工具将每个分词转换为 300 维的嵌入向量。对于未出现的词

（UNK）采用（-0.25, 0.25）区间的均匀分布进行随机初始化。由于后续的基模型需要统一的输入张量，需要对每个句子的长度进行填充（padding）或截断（truncate），本文使用训练集中最长的句子长度作为标准，根据统计为 42，对开发集和测试集也同样以这个长度作为填充（以 0 填充）或截断。

最后制作了为数据集的标签集，由于将 GoEmotions 分成了三个体系，一共要制作 36 种情绪对应的样本标签集。对于每个样本文本根据 GoEmotions 注释的标签，在相应 36 种情绪下给定 0 或 1，0 为不是该情绪或不属于该情绪的子类，1 为是该情绪或属于该情绪的子类。最终得到训练集，开发集，测试集的标签集。

4.1.3 基模型的选择与对比

本文采用的每个基模型的训练迭代数（epoch）是根据在开发集上的准确率和 loss 和训练集上的准确率与 loss 共同决定的，如图 4.2 以 Attention 模型在 Admiration 情绪上的训练迭代情况为例。随着 epoch 的增加，虽然训练集上的准确率和 loss 分别一直在上升和下降，但开发集上的准确率和 loss 在第 4 轮左右后分别开始下降与上升，明显发生了过拟合，因此选取的 epoch 为该转折点，即保证了高准确率又避免了过拟合。

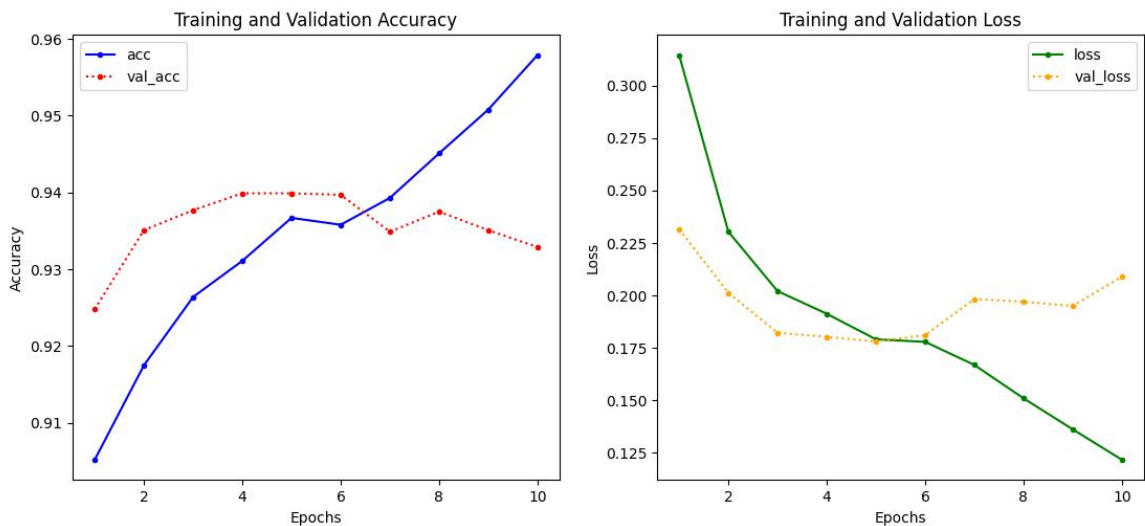


图 4.2 Attention 模型在 Admiration 情绪上迭代训练情况

接着，对所有基模型对 36 种情绪在测试集上的准确率进行了对比，选择最好的

基模型，如表 4.2，展示了部分五种基模型在测试集上对 36 种情绪的二分类效果。可以看到基模型在每个情绪上的准确率相差并不是很大，但 LSTM 总体上比其余 4 种基模型的准确率略微高，因此选用 LSTM 作为多层次模型的基模型。BiLSTM 模型并没有预期中比 LSTM 效果好很多，可能由于社交平台上的评论文本相对较短，没有充分发挥出双向的作用。此外，可能本文使用的 self-Attention 模型结构较简单，且句子较短，同样没有学习到更深的语义。BiLSTM+self-Attention 的组合模型，在部分情绪上确实比它们单个的模型准确率较高，但没有显著的增加。

表 4.2 五种基模型的分类效果

基模型	Positive	Ambiguous	Joy_ekman	Sadness_ekman	Surprise_ekman
LSTM	0.81113	0.88631	0.80155	0.94325	0.88502
CNN	0.80910	0.87931	0.80947	0.93606	0.88060
self-Attention	0.73945	0.88539	0.78017	0.93016	0.87525
BiLSTM	0.80726	0.88539	0.80818	0.93846	0.88428
BiLSTM+self-Attention	0.80818	0.87894	0.81279	0.94196	0.87857

表 4.2 五种基模型的分类效果（续）

基模型	Amusement	Admiration	Embarrassment	Annoyance	Realization
LSTM	0.96591	0.93164	0.99410	0.94214	0.97402
CNN	0.96130	0.92906	0.99392	0.94159	0.97494
self-Attention	0.96444	0.92685	0.99318	0.94104	0.97328
BiLSTM	0.96702	0.93072	0.99355	0.94085	0.97347
BiLSTM+self-Attention	0.96333	0.92758	0.99318	0.94104	0.97328

横向进行比较可以发现，第一层的三元情绪的分类效果没有第二层以及第三层的情绪高。可能是由于这三元情绪之间的情绪界限并不是很明显，导致较低的准确率。

4.1.4 实现过程

将 LSTM 的每个情绪对应的二分类模型训练后的网络结构保存为 json 文件，将权

重保存为 h5 文件。进行多层次模型分类时，使用每个情绪对应的二分类模型时再加载模型结构及权重。由于每个样本要加载的模型会很多，随着模型加载的积累会占用内存，导致预测的速度下降。为了解决该问题，每次加载完模型进行预测后立即清除掉 tensorflow 的 session。如下将该过程封装为函数：

```
def LoadModel(Mname,sample):
    with open(path + 'result_data\\' + Mname + '.json', 'r') as j:
        json_str = j.read()
    model = model_from_json(json_str)
    model.load_weights(path + 'result_data\\' + Mname + '_weights.h5')

    pro = model.predict(sample)[0][0]

    ## 清理内存
    K.clear_session()
    tf.compat.v1.reset_default_graph()
    return pro
```

最后将 36 种情绪的多层次模型，及采用最底层的 27 个情绪的二元分类模型的单层次模型在 5427 个文本的测试集上进行测试。

4.2 多层次模型改进实现

采用 pulp 包实现，整数线性规划，由于公式（3.1-3.6）所示，目标函数中含有绝对值，而 pulp 包中没有提供绝对值的处理。解决方法为，使用一个变量来表示目标函数的值，而用另一个变量来约束大于等于第一个变量的正负值。实现代码如下：

```
def ILP(P):
    emo_n = len(P)
    m = LpProblem(name='ILP', sense=LpMinimize)

    Y = [LpVariable(f'y_{i}', cat=LpBinary) for i in range(emo_n)]
    C = [(-np.log2(P[i]) + np.log2(1 - P[i])) for i in range(emo_n)]
```

```

## 目标函数

z = 0
for i in range(emo_n):
    z += Y[i] * C[i]

## 解决目标函数有绝对值问题

sum_var = LpVariable('sum_var')
abs_sum_var = LpVariable('abs_sum_var')

m += abs_sum_var
m += sum_var == z
m += abs_sum_var >= sum_var
m += abs_sum_var >= -sum_var
m += (lpDot([Y[i] for i in range(emo_n)], [1] * emo_n) >= 1)
m += (lpDot([Y[i] for i in range(emo_n)], [1] * emo_n) <= 3)

# 求解

m.solve()

res = [int(pulp.value(y)) for y in Y]
return res

```

根据前文的分析，空标签来自于第二层或第三层预测的概率值在 0.5 附近左右导致，因为从第二层开始使用 ILP 可能会影响到第三层的结果，为了对比从第二层开始使用 ILP 的第二层还是仅在第三层使用 ILP 效果好，分别在测试集上进行了测试。

4.3 实验结果与分析

在 5427 个样本的测试集上，分别测试了单层次模型（Baseline_Model），多层次模型（Multi_Level_Model），仅在第三层使用 ILP 改进的多层次模型（Multi_Level_ILP_Model），从第二层开始使用 ILP 改进的多层次模型

（Multi_Level_ILPs_Model）。以准确率与 F1 值对它们进行了比较。从表 4.3 可以看到，准确率和 F1 值上多层次模型比单层次模型都提高了约 0.05。使用 ILP 改进的多层次模型效果也比未改进的多层次模型得到了提高，仅在第三层使用 ILP 的模型相对原始多层次模型在准确率上提高了约 0.02，F1 上提高了约 0.03，而从第二层开始使用 ILP 的模型相对原始多层次模型在准确率和 F1 都上提高了约 0.05，与单层次相比提高了 0.1。可以看出本文提出的多层次模型方法和使用 ILP 改进的方法相对单层次模型有较好的效果。从表 4.4 可以看到，虽然单层次模型预测出的多标签数最多，但是准确率与 F1 值最低，说明预测出的多标签情绪误差大。

表 4.3 单层次模型与多层次模型，改进的多层次模型比较结果

模型	准确率	F1
Baseline_Model	0.38050	0.39478
Multi_Level_Model	0.43472	0.44725
Multi_Level_ILP_Model	0.45955	0.47364
Multi_Level_ILPs_Model	0.48285	0.49865

表 4.4 单层次模型与多层次模型，改进的多层次模型预测的多标签数(预测的情绪数大于 1)情况

模型	预测的多标签数（不区分分类对错）
Baseline_Model	242
Multi_Level_Model	148
Multi_Level_ILP_Model	133
Multi_Level_ILPs_Model	153

最后对多层次模型分类效果与单层次模型对比进行了具体的查看，对于句子“Enjoying Mummurs. I'm from Philly. I kind of terrified as they just used Skyrim music....”的真实标签为“joy”与“excitement”，单层次模型预测出的情绪标签为“joy”与“fear”，可以看出受到“terrified”词的影响，fear 对应的情绪分类模型很精准的将其识别出。而多层次模型，预测出的标签为“joy”一个。虽然只预测出了一个，但是并没有受到“terrified”词的影响，如图 4.3，在第一层三元分类时虽然稍微受到了“terrified”词影响，“Negative”情绪有 0.25 的预测概率但是并不是很强，因此并没有在最终预测出的标签中错误的引入“fear”标签。

```
Positive_LSTM: predicted-probability:0.856932520866394 predicted-label:1
Joy_ekman_LSTM: predicted-probability:0.5841723084449768 predicted-label:1
Joy_LSTM: predicted-probability:0.7120757699012756 predicted-label:1
Amusement_LSTM: predicted-probability:0.013321876525878906 predicted-label:0
Approval_LSTM: predicted-probability:0.06680551171302795 predicted-label:0
Excitement_LSTM: predicted-probability:0.0075750648975372314 predicted-label:0
Gratitude_LSTM: predicted-probability:0.023405522108078003 predicted-label:0
Love_LSTM: predicted-probability:0.003966540098190308 predicted-label:0
Optimism_LSTM: predicted-probability:0.0018102824687957764 predicted-label:0
Relief_LSTM: predicted-probability:0.0019007623195648193 predicted-label:0
Pride_LSTM: predicted-probability:0.0002938508987426758 predicted-label:0
Admiration_LSTM: predicted-probability:0.035711586475372314 predicted-label:0
Desire_LSTM: predicted-probability:0.011678129434585571 predicted-label:0
Caring_LSTM: predicted-probability:0.008320063352584839 predicted-label:0
Negative_LSTM: predicted-probability:0.25573796033859253 predicted-label:0
Ambiguous_LSTM: predicted-probability:0.02777770161628723 predicted-label:0
```

图 4.3 多层次模型预测具体情况

4.4 本章小结

本章对多层次模型与改进的多层次模型实现进行了具体的介绍。同时也介绍了实验环境，详细数据情况及预处理的方法，也将最终选择的基模型情况进行了对比与展现。最终比较了单层次模型，多层次模型，改进的多层次模型的实际实验结果，并进行了分析。

第 5 章 总结与展望

5.1 本文总结

本文以多标签情绪分类为问题导向，旨在面对情绪类别较多的复杂体系下，利用情绪类别的相关性从不同情绪的层次关系出发，建立多层次模型获得比单层次更好的分类效果。本文多标签分类的策略采用 BR 方法，将一个多标签分类任务建模为若干个单标签分类任务，使用基于深度学习的模型对每个层次的情绪构建了二元分类器。在分类器选择方面，进行实验分析对比了 CNN，LSTM，BiLSTM，self-Attention，BiLSTM+self-Attention 的 5 种常用的深度学习模型在同样情绪下的分类效果，选择效果最佳的模型作为基模型构建多层次模型。此外，为了最大程度发挥多层次的优势，对空标签情况，使用 ILP 对多层次模型进行了改进与优化。最后，设置了一系列对比实验，在 GoEmotions 数据集上，通过比较评价指标，验证了本文方法提出的有效性。

实验对比了常用于情感分类的 5 种深度学习模型，通过比较在相同情绪上的准确率，发现 LSTM 模型较优于其余模型，选择 LSTM 进行构建了多层次模型。准确率和 F1 值上多层次模型比单层次模型都提高了约 0.05。使用 ILP 改进的多层次模型效果也比未改进的多层次模型得到了提高，仅在第三层使用 ILP 的模型相对原始多层次模型在准确率上提高了约 0.02，F1 上提高了约 0.03，而从第二层开始使用 ILP 的模型相对原始多层次模型在准确率和 F1 都上提高了约 0.05，与单层次相比提高了 0.1。本文提出的多层次模型方法和使用 ILP 改进的方法相对单层次模型有较好的效果。

5.2 后期工作与展望

由于目前本文采用的多层次模型的情绪体系是人工划分的，且只试验了三元，ekman，GoEmotions 本身的 27+1 种情绪体系的组合，后期尝试使用其他情绪体系进行构建，或者探索出自动划分的方法。此外，由于使用的基模型相对简单，第一层的三元情绪的预测准确率并不是很高，后期使用更加复杂的深度学习模型进行提高。改进的多层次模型在第二层和第三层对预测出的概率值的阈值设定上，后期进行多此比

较使用最好的划分阈值，而不是目前阶段统一使用 0.65 作为 **cutoff**。此外，也是本文多层次模型的最大的限制在于，层次叠加的更多，包含的情绪越多，构建的二元分类模型也会大量增长，在训练基模型上非常费时费力，或许每个层可以使用一种损失函数对该层的误差统一进行计算，联合进行训练。

参考文献

- [1] Safa R, Bayat P, Moghtader L. Automatic detection of depression symptoms in twitter using multimodal analysis[J]. The Journal of Supercomputing, 2022, 78(4): 4709-4744.
- [2] Pla F, Hurtado L F. Political tendency identification in twitter using sentiment analysis techniques[C]//Proceedings of COLING 2014, the 25th international conference on computational linguistics: Technical Papers. 2014: 183-192.
- [3] Desai J, Cao H, Shah R. Attention-based Region of Interest (ROI) Detection for Speech Emotion Recognition[J]. arXiv preprint arXiv:2203.03428, 2022.
- [4] Plutchik R. Emotion[J]. A psychoevolutionary synthesis, 1980.
- [5] Ekman P. An argument for basic emotions[J]. Cognition & emotion, 1992, 6(3-4): 169-200.
- [6] Russell J A, Mehrabian A. Evidence for a three-factor theory of emotions[J]. Journal of research in Personality, 1977, 11(3): 273-294.
- [7] Godbole S, Sarawagi S. Discriminative methods for multi-labeled classification[C]//Advances in Knowledge Discovery and Data Mining: 8th Pacific-Asia Conference, PAKDD 2004, Sydney, Australia, May 26-28, 2004. Proceedings 8. Springer Berlin Heidelberg, 2004: 22-30.
- [8] Read J, Pfahringer B, Holmes G, et al. Classifier chains for multi-label classification[J]. Machine learning, 2011, 85: 333-359.
- [9] Tsoumakas G, Katakis I, Vlahavas I. Random k-labelsets for multilabel classification[J]. IEEE transactions on knowledge and data engineering, 2010, 23(7): 1079-1089.
- [10] Farruque N, Huang C, Zaiane O, et al. Basic and Depression Specific Emotions Identification in Tweets: Multi-label Classification Experiments[C]//Computational Linguistics and Intelligent Text Processing: 20th International Conference, CICLing 2019, La Rochelle, France, April 7 - 13, 2019, Revised Selected Papers, Part II. Cham: Springer Nature Switzerland, 2023: 293-306.
- [11] He H, Xia R. Joint binary neural network for multi-label learning with applica

- tions to emotion classification[C]//Natural Language Processing and Chinese Computing: 7th CCF International Conference, NLPCC 2018, Hohhot, China, August 26 – 30, 2018, Proceedings, Part I 7. Springer International Publishing, 2018: 250-259.
- [12] Kim Y, Lee H, Jung K. Attnconvnet at semeval-2018 task 1: Attention-based convolutional neural networks for multi-label emotion classification[J]. arXiv preprint arXiv:1804.00831, 2018.
- [13] Jabreel M, Moreno A. A deep learning-based approach for multi-label emotion classification in tweets[J]. Applied Sciences, 2019, 9(6): 1123.
- [14] Baziotis C, Athanasiou N, Chronopoulou A, et al. Ntua-slp at semeval-2018 task 1: Predicting affective content in tweets with deep attentive rnns and transfer learning[J]. arXiv preprint arXiv:1804.06658, 2018.
- [15] Demszky D, Movshovitz-Attias D, Ko J, et al. GoEmotions: A dataset of fine-grained emotions[J]. arXiv preprint arXiv:2005.00547, 2020.
- [16] Shaw P, Uszkoreit J, Vaswani A. Self-attention with relative position representations[J]. arXiv preprint arXiv:1803.02155, 2018.
- [17] Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space[J]. arXiv preprint arXiv:1301.3781, 2013.
- [18] Pennington J, Socher R, Manning C D. Glove: Global vectors for word representation[C]//Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 2014: 1532-1543.
- [19] Joulin A, Grave E, Bojanowski P, et al. Bag of tricks for efficient text classification[J]. arXiv preprint arXiv:1607.01759, 2016.
- [20] Zhu S, Li S, Chen Y, et al. Corpus fusion for emotion classification[C]//Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers. 2016: 3287-3297.

致 谢

感谢朱苏阳老师在毕设上的悉心指导以及改进意见，无论何时都能快速回答我的各种问题和提供了宝贵的思路。感谢李秉千学长提供的硬件环境，使得我的模型训练时间大大减少。感谢学校提供的计算机辅修这一机会，使得我能朝着我热爱的方向又迈了一步。