基于多步决策的文本情感识别

（申请清华大学工学硕士学位论文）

培 养 单 位：计 算 机 科 学 与 技 术 系
学　　科：计 算 机 科 学 与 技 术
研　究　生：梁　锡　豪
指 导 教 师：徐　明　星　副　教　授

二〇一九年五月

Multi-step-decision-making Based Text Sentiment Analysis

Thesis Submitted to

Tsinghua University

in partial fulfillment of the requirement

for the professional degree of

Doctor of Engineering

by

Liang Xihao

(Computer Science and Technology)

Thesis Supervisor : Professor Xu Mingxing

May, 2019

关于学位论文使用授权的说明

本人完全了解清华大学有关保留、使用学位论文的规定，即：

清华大学拥有在著作权法规定范围内学位论文的使用权，其中包括：（1）已获学位的研究生必须按学校规定提交学位论文，学校可以采用影印、缩印或其他复制手段保存研究生上交的学位论文；（2）为教学和科研目的，学校可以将公开的学位论文作为资料在图书馆、资料室等场所供校内师生阅读，或在校园网上供校内师生浏览部分内容。

本人保证遵守上述规定。

（保密的论文在解密后应遵守此规定）

作者签名：_____　　导师签名：_____

日　　期：_____　　日　　期：_____

摘 要

自 Web2.0 普及后，人们逐渐习惯在互联网的各种平台上分享他们的想法和情感。透过对这些媒体进行情感分析，我们可以得知人们对特定人事物的想法和态度。对人们的想法快速作出响应能够带来相应的商业价值和政治价值，其相关技术也因此得到了重视。而文本作为社交平台上的主要媒体之一，面向文本的情感识别在近年也成为了热门的研究领域。本论文探讨了文本情感识别中的两个问题，主要内容如下：

1. **基于多步决策的多分类系统框架**。随着现实中应用场景变得复杂，需要解决的多分类问题越来越多。当区分的类别越多，机器学习算法对数据进行拟合的难度更高，另外对识别性能的要求也变得复杂，譬如确保个别类别的召回率和正确率等。为此，我们提出了一种基于多步决策的多分类系统框架，把一个多分类问题拆解成多个子分类问题的叠加，再透过逐步回答每个子问题得出最终的识别结果。由于识别过程中每一步只关注一个子分类问题，我们能从局部调整系统的识别能力，并针对特定的评价指标进行优化。另一方面，每个子分类问题可以分别采用不同的算法，以此结合不同模型能捕捉到的不同信息。
2. **用于结合上下文的多通道模型**。在某些场景下，仅凭一段文本无法准确了解发言者想表达的意思和态度。为了处理这种情况，一些文本情感识别研究会引入上下文作为提示。为此我们提出了一种多通道模型框架，让对识别目标起不同作用的上下文先分别经过不同的编码器进行编码提取出有用的特征，再考虑合并正文和上下文的信息得出识别结果。

为了验证基于多步决策的多分类系统框架，我们首先将其应用于面向微博的反讽识别，根据国际比赛 SemEval-2018 的任务三进行实验，结果显示我们的模型超过了当时排名第一的系统。另外为了验证我们提出的多通道分类模型，我们将其应用于面向三轮对话的情感识别，根据公开比赛 SemEval-2019 的任务三进行实验，结果显示我们的模型达到了当时排名前十的性能。

关键词：情感分析；反讽识别；深度学习，集成学习

Abstract

Since the poularization of Web 2.0, people get used to share they thoughts and emotions on different online platforms. By analyzing the sentiment of these data, we can have an in sight into the people's opinion towards certain things. Vast commercial and political values can be obtained through quick response to the people's attitude. Therefore related technologies are getting more attention. As text is one of the mostly used media on social platforms, text sentiment analysis has become a popular research field. This thesis explores two problems in text sentiment analysis and it is organized as follows:

1. **Multi-step-decision-making based multi-classification system framework**. As real-world application scenes are becoming more complex, more multi-classification problems are encountered. As the data are divided into more categories, it becomes more difficult for machine learning algorithms to fit the data. On the other hand, the requirement of predictive performance are getting more complicated, such as ensuring the recall rate and the precision level of certain categories. Hence, we propose a multi-step-decision-making based multi-classification system framework, which disassembles a multi-classification problem into a sequence of sub-classification problems and classifies an object by answering the sub-questions step by step. As the classification system only focuses on a sub-problem at each step, its performance can be adjusted locally and we can optimize the overall performance for certain evaluation metrics. On the other hand, algorithms can be chosen for each sub-problem individually, hence the information captured by different kinds of models can be combined in the system.
2. **Multi-channel model for manipulating contextual information**. Sometimes we may not exactly understand what a person want to express or how he feels only through his written message. To deal with this problem, some of the text sentiment analysis researches take into account the contextual clues. Therefore, we propose a multi-channel model. By feeding contextual information that play different parts of role into different feature encoders, features of different parts of the context are encoded and further combined with that of the main message to get the identification result.

To verify the effectiveness of the proposed multi-step-decision-making based multi-

classification system framework, we first apply it to irony detection in English tweets. Experiments are launched based on SemEval-2018 Task 3. Results show that our system exceeds the performance of the no. 1 participating system. On the other hand, to verify the effectiveness of our multi-channel model, we apply it to contextual emotion detection in text. Experiments are launched based on SemEval-2019 Task 3. Results show that our system reaches the top 10 performance among the participants.

Key words: Sentiment analysis; irony detection; deep learning; ensemble learning

目 录

主要符号对照表

BOW	词袋模型 (Bag of Words)
BP	反向传播算法 (Back-propagation)
BRNN	双向递归神经网络 (Bidirectional Recurrent Neural Network)
CRF	条件随机场 (Conditional Random Field)
GRU	门控循环神经元 (Gated Recurrent Unit)
LSTM	长短时记忆网络 (Long Short-Term Memory)
POS	词性 (Part-of-speech)
ReLU	线性整流函数 (Rectified Linear Unit)
RNN	递归神经网络 (Recurrent Neural Network)
SVD	奇异值分解 (Singular Value Decomposition)
SVM	高性能计算 (High Performance Computing)
TF-IDF	词频-逆文档频度 (Term Frequency - Inverse Document Frequency)

第 1 章　引言

1.1　研究背景与意义

情感识别旨在了解人们对特定事件或实体的态度和情感。自 Web2.0 普及后，大量网民每天在互联网生产着各种各样的内容，其中包含不同方面的信息，如个人生活经历，购买行为，对产品服务的体验评价，对社会时事的看法等等。从人们的日常社交需求来看，这种借由互联网媒体的分享非常便捷，我们可以了解到亲朋好友的近况，也可以和不认识的网民交流对具体事件的想法。在商业上，借由对用户的网络行为进行分析，企业可以对他们的客户或者潜在客户有更深入的了解，对他们的需求和反馈及时作出反应将带来战略性的优势。在社会管理上，政府可以透过对网民在网络上的发言了解人民的想法和舆论的走向，进而作出相应的措施。正因为互联网的普及，才使得以上基于对特定人群的了解来进行决策的做法成为可能。随着数据资源变得丰富，相应的技术在近年得以快速发展，目前市面上已经有公司（如国内的腾讯和阿里巴巴，以及国外的微软和亚马逊等）提供基于大型社交媒体平台 (如微博、讨论区等) 上的数据进行情感分析相关的规泛化服务，然而相应的技术依然有进步空间，研究工作还在不同方向上摸索。

情感在人们的思想表达和交流中起着重要作用[1]，比起了解该想法的细节内容，情感对应该想法的一种倾向。譬如在分析用户对新产品的评论时，只需要分析其中褒贬意思的倾向即可大致了解新产品是否能让大部分的客户满意，或者筛选出其中表示不满意的用户再进行深入分析，因此情感识别技术具有一定的应用价值。而由于在互联网上，大部分情况下用户以文本表达想法，面向文本的情感识别成为了近年最重要的研究课题之一。相对于人们面对面交流的场景，聆听者可以根据发言者的肢体语言、面部表情以及声调变化等额外信息更好地理解发言者想表达的意思，然而这些信息并不存在于文本当中，这也正是对文本进行情感识别本身的难点之一[2]。

情感识别的另一个难点在于语言中丰富多样的修辞手法，其中反讽是具代表性的修辞手法之一。Henry Watson Fowler 在《The King's English》一书中描述“即使对反讽的定义有数百种, 其中只有包含' 表面意思和实际意思不同' 这个概念的才能被接受”。Eric Partridge 在《Usage and Abusage》一书中指出“反讽存在于所表达意思的另一面”。总的多说当反讽在文本中出现，那么发言者想表达的意思应该和文本的字面意思完全相反。譬如某人表示“我就喜欢你不断挑战我的底线”，在字面上“喜欢”表达的是正面的情感，然而根据常识可知“挑战底线”是一种让人反感的

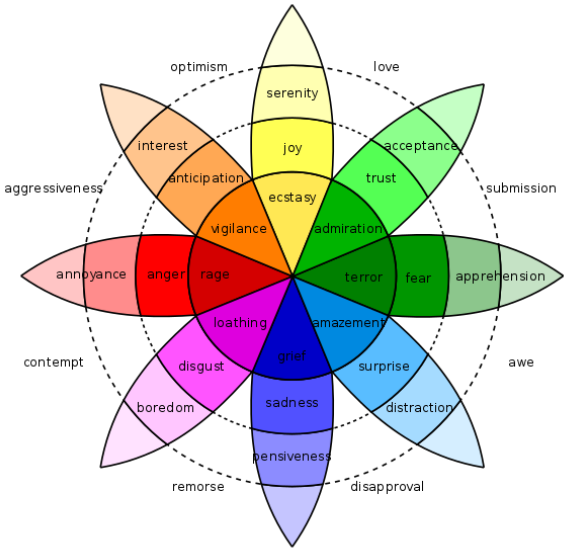
行为，与“喜欢”相矛盾。这段文字实则表示发言者正被某人触及底线，表达的是负面情感。所以在情感识别当中，正确识别出反讽的使用能够避免对内容的错误理解，反讽识别因此成为了和情感识别紧密相关的研究课题。

1.2　国内外研究现状

1.2.1　情感模型

情感计算的基础是对情感作出描述，现有的描述方式可以分成两个大类: 范畴观和维度观。范畴观即把不同情感对应到一组离散的情感标签上，其中具代表性的有 Plutchik 的情感模型[4] 和 Ekan 的情感理论[5]。Plutchik 的情感模型包含八种情感：愤怒、恐惧、悲伤、厌恶、期待、信任、高兴、惊讶；这些情感都各自对应具有重要生存意义的行为，各种复杂的情感都是由这此基本情感构成，另外这八种情感可以分成四组对立的情感对。Ekan 在十年后提出的情感理论和 Plutchik 的相似，但相对地少了期待和信任两种情感，是一个六类情感模型。

(a) 三维模型



(b) 二维模型

图 1.1　Plutchik[4] 提出的情感轮模型

以维度观描述情感就是把情感映射到多维空间的点上，而目前维度观情感模型以二维和三维空间的为主。二维情感模型中较有代表性的是 Russell 提出的环状模型（Circumplex Model)[6] ，其中纵坐标对应情感的激活度（Arousal)，横坐标对应情感向性（Valence)，而不同的情感则分布在一个环状的区域内。Bradley 等人[7] 提出的向量模型（Vector Model）在对横轴和纵轴的定义和前者相似，但其理论假

设高激活度的情感应该有较明显的正负倾向，相对地低激活度的情感则偏向中性，故情感分布在一个回力标形状的区域。Watson 和 Tellegen[8] 提出的 PANA（positive activiation-negative activation）模型和前两者在理论基础上则有明显的不同，他们认为情感的正面作面和负面作用是两个独立的成分，所以在模型中纵轴和横轴分别表示情感正面作用和负面作用的强弱，不过该模型相当于把 Russell 等人提出的环状模型的向量空间旋转 45 度[9]。

基于三维空间的情感模型中具有代表性的有 Plutchik[4] 提出的情感轮模型，Plutchik 认为情绪之间包含强度，相似性和两极性三种维度，椎体的顶部和底部分别对应强的情感和弱的情感，相似的情感对应椎体中相近的位置，对立的的情绪则会对应到椎体中对立的位置上。另外还有 Mehrabian[10] 提出的 PAD 模型，其三维空间的三个坐标轴分别对应情感愉悦度（Pleasure）、激活度（Arousal）以及优势度（Dominance）。较近期被提出的是 Hugo[11] 的情感立方体模型，其三维空间的三个坐标轴分别对应 5-羟色胺（5-hydroxytryptamine, 5-HT），多巴胺（dopamine，DA）和去甲肾上腺素（noradrenaline，NE）三种神经递质所产生信号的强弱，并对空间中一个立方体的八个顶点标记了其对应的情感。

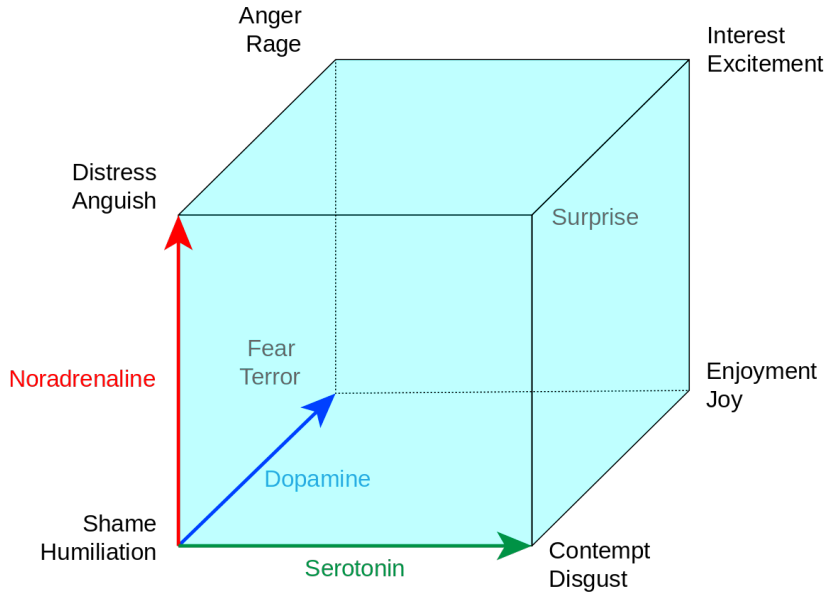


图 1.2　Hugo[11] 提出的情感立方体模型

1.2.2　情感识别

对应上述情感模型的分类，情感识别研究可以分成两类。第一类对应范畴观，给定一组情感类型，判断一段文本中所表达的情感倾向于该组情感中的哪一种，或者是否包含这一组情感中的一种或多种情感。如国际比赛 SemEval-2018 任务一[12]

的子任务要求识别一段微博中是否包含愤怒、恐惧、悲伤等十一种情感中的一种或多种情感。另一类情感识别研究对应维度观，对于给定的情感属性，判断一段文本中该情感属性的强度。其中常见的有情感向性的二分类问题（正性或负性）、三分类问题（正性、中性或负性）、五分类问题（非常正性，正性、中性、负性或非常负性）。其中五分类问题的研究对象一般是互联网上五星评分制的产品评论或者电影评论等。

另一方面，目前文本情感识别的研究按照文本的粒度可以大致分成三类：文章级别、句子级别、属性级别。在文章级别的情感识别中，虽然一篇文章由多个句子组成，但假设整篇文章存在某种情感倾向，而研究目标则是自动识别出该种情感倾向的类型或强度。Turney[13] 利用线上电影评论中" 推荐"（大拇指朝上）和" 不推荐"（大拇指朝下）的标记研究对电影评论的正负性情感识别。他提出利用点互信息 (Pointwise Mutual Information，PMI) 来评估单词的语义倾向性，其中 PMI 透过在大型语料库中统计两个单词共同出现的情况来评估两个单词的相似度。首先利用 PMI 值评估各个单词与正向情感的代表性单词"excellent" 和负向情感的代表性单词"poor" 的相似性，再取这两个 PMI 值作差得出该单词在语义上倾向于哪一种情感，再透过计算整段评论的平均语义倾向性评估整体的情感倾向。Pang 和 Lee[14] 采用了不同的方法研究相同的问题。他们首先对评论中每个句子的主观程度进行评分，利用评分构造一个带权重的句子关系图，再基于最小割算法结合上下文加强对每个句子主观程度的判断，过滤评论中不带主观情感的句子后再判断整个评论的情感向性，以此加强识别效果。Tang 等人[15] 则研究了产品评论的五级评分预测。除了评论的文本内容，他们进一步引入了用户的信息和产品的信息。经过数据分析，他们发现相同用户对不同产品的评论和评分比不同用户之间的更一致，另外不同用户对同一产品的评论和评分比不同产品之间的更一致，这显示了用户和产品各自都存在一些相对固定的属性。因此 Tang 等人提出一种为用户和产品生成表示向量的方法，并应用于评论的五级评分预测。实验结果显示他们的方法在多个数据集上达到了较好的性能，这同时引出了加入背景信息来加强系统识别能力的可能性。

然而一篇文章有可能同时表达了多种观点和情感，因此有另一类研究针对句子或短文本所表达的情感，即句子级别的情感识别。由于句子的文本长度较文章的短，文本内部的逻辑较简单，但相对地所包含的提示信息也较少，课题的难点与前者有所不同。Khan 等人[16] 研究了对线上评论中句子进行正负性情感识别。他们首先区分出评论中各个句子的主观性和客观性，然后针对带主观情感的句子，利用开源自然语言工具 SentiWordNet 获取各个单词的正负情感属性，再根据句子的

词性标注和他们设计的规则计算整个句子的情感向性。Khan 等人默认单词的正负情感属性在不同场景下不变，然而 Li 等人[17] 指出部分单词在特定场景下会有不同的情感向性，因此他们提出了一种有监督学习方法，自动学习各单词在指定领域下的情感向性，以此作为文本的特征，并应用于对产品评论的情感识别中。实验结果显示使用 SentiWordNet 提供的全局情感评分和针对领域评估的情感评分相比，使用后者能达到的识别性能力更好，验证了他们的假设。一些研究则选择端到端地学习单词的情感和语义倾向，在 Santos 和 Gatti[18] 对电影评论和微博的情感识别研究中，他们提出了利用卷积神经网络分别从字符级别和词级别计算出单词对应的特征向量，然后同样以卷积神经网络结合句子中各单词的向量得出句子的表示向量和预测整体的情感倾向。结果显示他们的方法较早期的其他方法性能更好。

在一些应用场景当中，我们希望了解发言者对某个对象或者它的某个特定属性的想法。譬如新手机推出市场后，厂商需要了解用户对手机的续航能力、拍照质量、交互体验等各方面的评价，那么对于评论中同时谈论手机的多个属性且好评和差评不一时，应该针对各个属性分别识别发言者所表达的感情，因此有了属性级别的情感识别。Che 等人[19] 提出一种句子压缩算法，透过对句子进行依存句法分析，过滤与目标属性的情感无关的内容。他们采用了多种语义和语法特征作为输入，以条件随机场 (Conditional random field，CRF）作为分类器。实验结果显示过滤掉不相关的文本部分后，识别性能有所提升。Wang 等人[20] 研究了对网上评论中特定实体或属性的情感识别。他们提出了一个基于注意力机制的长短时记忆网络（Long Short-Term Memory，LSTM），特点在于只以词向量作为输入，而不采用其他传统的语义和语法特征，另外利用注意力机制自动识别与目标相关的内容。他们的实验基于国际比赛 SemEval-2014 任务四[21] 中的一个子任务，结果然显示他们的系统性能达到了当时的技术水平。另外透过对注意力单元的输出进行分析，验证了注意力机制能够有效识别文本中与目标相关的内容。Tang 等人[22] 分别对前述数据集中电脑和餐厅相关的两组样本进行属性级别的情感识别，但有别于当时主流的以特征提取为主的浅层机器学习方法（如支持向量机）和针对序列的深度学习模型（如递归神经网络），他们提出了一个基于注意力机制的深度记忆网络。另外针对文本中各个单词和目标单词在句子中的距离，作者引入了距离信息提出了注意力单元的多种变形。实验结果显示他们提出的深度记忆网络达到了当时第一名参赛系统（基于手工特征和支持向量机的算法）的性能。另外对注意力单元的中间结果进行人工分析，验证了在同一段评论中，引入距离信息的注意力单元有助于区分不同单词对不同目标的情感的贡献。

1.2.3　反讽识别

正如前面所述，反讽识别和情感识别紧密相关，识别出反讽的使用对正确识别情感起着关键作用。Tsur 等人[23][24] 研究了对微博平台 Twitter 上的微博以及电商平台亚马逊上的评论进行反讽强度的识别，从明显不含反讽到明显表示反讽分成五级，由人工进行标注。他们提出的 SASI 算法分别从文本提取了词频相关的模式特征以及基于标点符号的特征，以 K 最近邻算法作为分类器。另外利用在 Twitter 按井号标签 #sarcastic 自动爬取了额外的反讽语料用于初步的模型训练。对实验结果的比较证明了各种特征的有效性以及添加额外语料对模型训练的帮助。

为了以较低成本获取大量的反讽相关语料，很多研究从微博平台根据井号标签自动筛选出可能带反讽的微博。Reyes 等人[25] 利用 #irony，#education, #humor, #politics 在 Twitter 上自动获取四组不同主题的微博，并把 #irony 对应的微博和另外三组微博两两组合进行二分类实验。他们提出了四个方面的文本特征提取方法，包括：特殊标记（词汇和标点符号等）、不可预期性、表达风格、情感特性；并比较了每项特征在各组微博中的出现情况，显示了各项特征和反讽的相关性。另外分别采用朴素贝叶斯和决策树作为分类器，但没有明显的性能区别。

类似的数据收集方法对其他语言同样适用。Kunneman 等人[26] 则利用对应的德语井号标签来获取反讽语料，以此研究德语微博中的反讽识别。他们取 N 元语法作为输入特征，以 Balanced Winnow[27] 作为分类器，在测试集上达到约 0.85 的召回率和 0.87 的 AUC 值。但他们进一步经过人工检验评估了基于井号标签自动标注的有效性，分析结果显示该方法获取的反讽样本中包含约 10% 的躁声。

有别于早期以手动设计的特征作为输入和以传统机器学习方法建模，Poria 等人[28] 首次将神经网络应用于对微博的反讽识别。他们的算法框架主要包含四个卷积神经网络，并分别利用不同的数据集进行预训练，分别对应反讽识别、情感极性识别、情感类型识别和性格识别。最后取四个卷积神经网络的中间结果作为特征，利用支持向量机 (Support Vector Machine, SVM) 进行后融后得出最终预测结果。实验显示引入反讽识别以外的三个语料库提升了系统的识别能力。

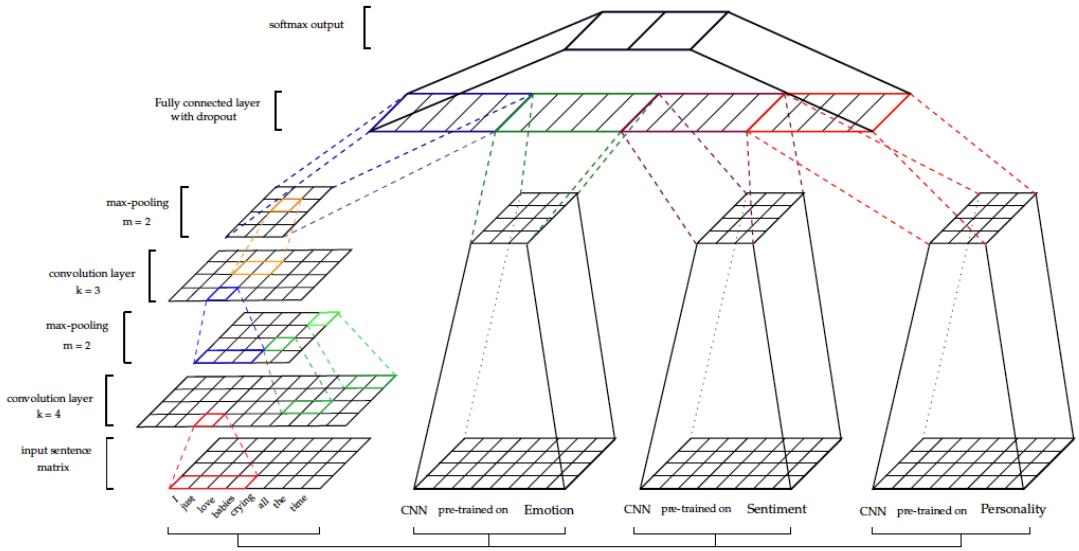


图 1.3　引入多个领域信息的反讽识别神经网络模型，引自 [28]

1.2.4　社交媒体上的文本情感识别

随着社交媒体的普及，网民们习惯于在微博、讨论区等平台上表达个人意见和互相讨论。有别于新闻或学术材料等较正式的文件，网民可以较随心所欲地发言，社交媒体因此成为了情感识别的重要研究对象之一。但和产品评论或文章等文本类型不同，社交媒体上的文本普遍较短[29]，缺少对背景信息的提示，难以判断其内容所属的主题，这对正确理解其中表达的意思带来困难。另外语言中夹杂着不正规的用法，在中文微博中会出现新的短语或对旧短语有新的解释[30]，如“锦鲤”暗示“好运”，“灌水”表示发表没有意义的内容，在英文微博则中会出现错拼字、非正式缩略语、表情符[31][32]，如“tnx”对应英文单词“thanks”，“:) ”表示微笑等。虽然没有正式的语言组织对这些新的用法进行整合，但因为这些新用法更方便或对思想的表达更丰富到位，随着在网络上的传播而在网民之间达成了共识。传统的文本特征提取建立在标准的单词使用和正规的语法结构上，而由于新词汇和新语用的出现，导致词汇的意思无法被识别或被错误理解，这对于由人工智能理解文本内容成为了一大难点。因此有别于传统的文本研究，在面向社交媒体的文本情感识别时，需要采用额外手段对文本进行预处理，或透过大量语料尝试自动学习这些新出现的语言属性。

Khan 等人[33] 研究了微博的正负中性情感识别。他们提出了一个混合三个分类器的情感识别框架来解决数据稀疏的问题，另外还提出了一组针对微博文本的预处理步骤，其中包括俚语和缩略语分析、词干提取、拼写检查和修正、用户名和井号标签移除等。他们的系统在 6 个微博数据集上达到了平均 83.3% 的 F1 值以及

平均 85.7% 的准确率，和同类型技术比较后验证了他们系统以及预处理手段的有效性。

Angiani 等人[34] 针对英语微博比较了各种常用的文本预处理技术对情感分析的影响，其中包括单词的规范化、表情符到情感标签的映射、俚语映射、词干提取、停词过滤等。分析显示除了俚语映射以外，其他技术均对情感识别都有正面影响，其中部分技术有助于统一拼写相似的单词，以此关联相同概念的词组。但同时采用所以预处理技术并不保证达到最好的效果，作者指出依然需要根据应用场景和文本的特性做选择。

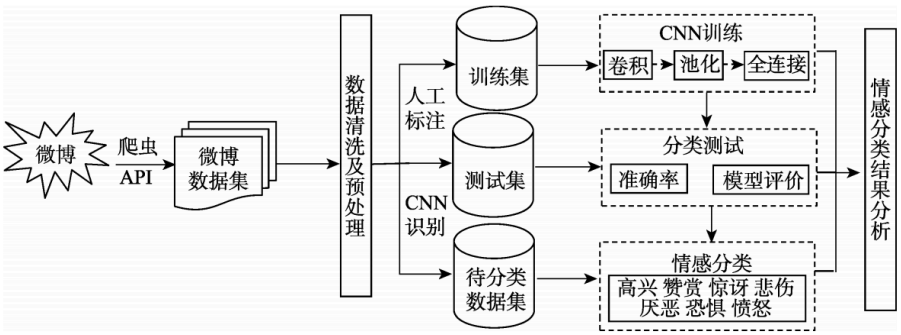


图 1.4　基于卷积神经网络的微博情感分类模型，引自[35]

张海涛等人[35] 则研究了中文微博和评论中的文本情感分类。他们针对当时微博上具一定争议性的话题 # 打呼噜被室友群殴 # 收集数据，确保了样本围绕同一个主题并且有充足的数据量。使用开源工具 NLPIR/ICTCLAS2016 对语料进行分词，再以词向量学习算法 word2vec 从语料中学习词的表示向量作为输入，以基于卷积神经网络的模型作为分类器，另外以支持向量机作为对照算法。实验结果显示在面向长文本的情感识别时，他们的系统比支持向量机性能更好，而面向短文本时则相反。

1.3　存在的问题

由于近年来各企业或机构对情感识别的需求增加，相关技术的研究备受关注，加上深度学习的快速发展，情感识别和反讽识别的性能水平也在逐年提升。然而相关研究依然存在一些问题需要深入探讨：

- **数据不均匀在多分类问题中的影响**。情感识别在早期以识别文本的正性、负性和中性情感为主，但随着应用场景越来越复杂，相关研究逐渐关注其中的细分类别，反讽识别等研究也如此。而在多分类问题中，各类别样本量分布不均匀一直是个备受关注的问题。因为在真实应用场景的多分类问题中，各

个类别的样本量往往分布不均匀甚至差别很大，这会导致部分性能指标明显偏低。譬如在训练数据中样本量较少的类别在测试集上的召回率会明显比其他类别的低，间接拉低宏平均的召回率和 F1 值。然而目前主流的研究工作大部分只探索单个模型如何对多类别的数据进行建模，在人工神经网络领域则是不断提出新的网络结构以在整体上达得到更好的识别能力。另外也有透过数据增强的方法调整训练数据中样本的分布情况，但也难以针对个别类别的识别能力进行调整。

- **在算法建模中引入上下文信息** 在一些场景下，仅凭一段文本的内容可能无法完全理解发言者想表达的内容，这在短文本的场景中尤其明显，如微博、评论等。为此目前一些研究会考虑引入文本的上下文，认为这些上下文中包含相关的信息，有助于理解原文的内容。但在不同场景下，上下文的类型不同，其对应建模方式也会有所不同。如在微博平台中，要对微博正文进行情感识别，那么可以引入微博底下的评论以定位该微博描述的事情，另一方面也可以引入用户过去发布的微博，对比他使用表情符的习惯和当前微博中使用的表情符来猜测用户想表达的情感。然而对于不同场景下不同类型的上下文，如何在算法建模中引入上下文信息始终没有一种通用的方法，对于具体的问题依然需要进行针对性的设计。

1.4　本论文的内容安排

本文针对上述的两个问题，提出了一个基于多步决策的多分类系统框架以及用于结合上下文的多通道分类模型，并分别基于两个应用场景进行实验以验证他们的有效性。本论文的内容安排如下。

在第2章，我们会对面向文本的情感识别问题作出分析，给出一个统一的数学定义。然后我们会给出一个文本情感识别的研究框架，详细说明其中每一步完成的任务和目的。紧接着会介绍一些自然语言处理和情感识别的相关技术，对于后续实验用中使用到的技术，我们会给出相对充分的说明，另外也会给出相关技术的概要描述，以便于其他研究者在本论文未深入探索的方向进行拓展性的工作。

第3章中，我们会首先介绍一种基于多步决策的多分类系统框架，并应用于面向微博的反讽识别。我们将基于国际比赛 SemEval-2018 的任务三[36] 展开实验，其中包含两个子任务。子任务一为二分类问题，要求识别微博是否包含反讽的修辞手法。另一个子任务为四分类问题，数据与子任务一相同，但原本" 反讽" 一类的样本被细分成反讽的三个类型：基于相反语义的反讽、情景反讽、其他反讽。重点地针对子任务二，基于我们提出的多分类系统框架，我们给出了一个具体的基于

多步决策的反讽识别系统。我们会对系统中各个部分进行对比实验，另外会对系统的最终识别结果和中间结果进行分析，以此验证我们提出的系统框架的有效性以及解释其中的合理性。最后进行错误分析，了解我们系统在此研究问题中的不足之处。

第4章会介绍一种多通道分类模型，用于结合在情感识别中起着不同作用的上下文信息。我们将把该模型应用于面向三轮对话的情感识别，并基于国际比赛SemEval-2019 的任务三[2] 展开实验，该比赛要求参赛系统识别两人轮流发言的三轮对话中最后一轮发言者所表达的情感，一共涉及四个情感类别：高兴、悲伤、愤怒、其他。同样地，基于我们提出的多分类系统框架，我们给出了一个具体的基于多步决策的情感识别系统，透过和参赛系统的性能进行比较评估我们系统的性能水平。另外，我们依然会对系统中各个部分进行对比实验和分析，以进一步说明在特定数据分布下，如何针对特定评价指标设计基于多步决策的多分类系统。

最后，在第5章中，我们将总结本篇论文中的主要贡献和实验结论，并根据前面各实验的错误分析为后续研究工作给出建议。

第 2 章　问题分析与研究框架

2.1　本章引论

面向文本的情感识别和相关研究多种多样，如前一章中提到的面向评论的五星评分预测、面向微博的正负中性情感识别、面向评论的五级反讽强度识别等。虽然这些课题关注的文本特征不同，需要区分的类别不同，采用的数据处理技术和算法也因此有所不同，但它们在本质上有相同之处，其研究方法大同小异。

本章的内容安排如下。在章节 2.2中，我们将首先分析情感识别的分类问题所涉及的要素，并给出统一的形式化表示。基于该形式化表示，在章节 2.3中，我们会提出一个面向文本的情感识别研究框架，说明每个步骤中需要实现的功能和目的，在后续章节中，我们将基于此研究框架进行我们的工作。紧接着在章节 2.4中，我们会根据研究框架中需要实现的各种功能，介绍其相对应的技术。

2.2　问题分析

本小节中，我们将对面向文本的情感识别所涉及的各项要素作出分析，并给出统一的形式化表示，以描述这些要素之间的关系。

在情感识别的分类问题中，我们需要把情感区分为有限个不同的类别，这些类别组成情感集合 C。如 Tang 等人[15] 的情感极性识别研究，C 对应五个级别的情感。在刘丹丹等人[37] 的微博情感分类研究中，C 对应情感类别为喜好、安乐、惊奇、厌恶、悲哀、愤恨、恐惧。另外在反讽识别问题中也同理，在邓钊等人[38] 的中文反语识别研究中，C 则对应包含反讽和不包含反讽这两个类别。

而在面向文本的情感识别问题中，必然有文本内容 T，它由情感持有者 S 发表，是他表达个人想法和情感的载体。在一些场景下还会有上下文 B，对应和 T 相关的内容，有助于了解 T 表达的意思。譬如在研究讨论区上帖子的情感类别时，T 对应帖子的文本内容，S 则是帖子的发布者。而帖子所在讨论区的类型有助于定位帖子对应的主题，发布者的发布历史可能隐藏发布者的一些态度倾向，帖子下的评论从侧表反映主帖的内容，这些都可以作为上下文 B。又以 Zahiri 等人[39] 对电视剧台词的情感识别研究为例，T 对应电视剧中的台词，S 为说台词的对应角色，其他角色的台词可以作为上下文 B。

那么对于任意一段文本 $t \in T$，在给定上下文 $b \in B$ 的情况下，情感识别假设 t 表达的情感必然对应某种情感类别 $c \in C$。我们的目标是找出一个映射关系 F_C，

使得 $c = F_C(t, b)$。

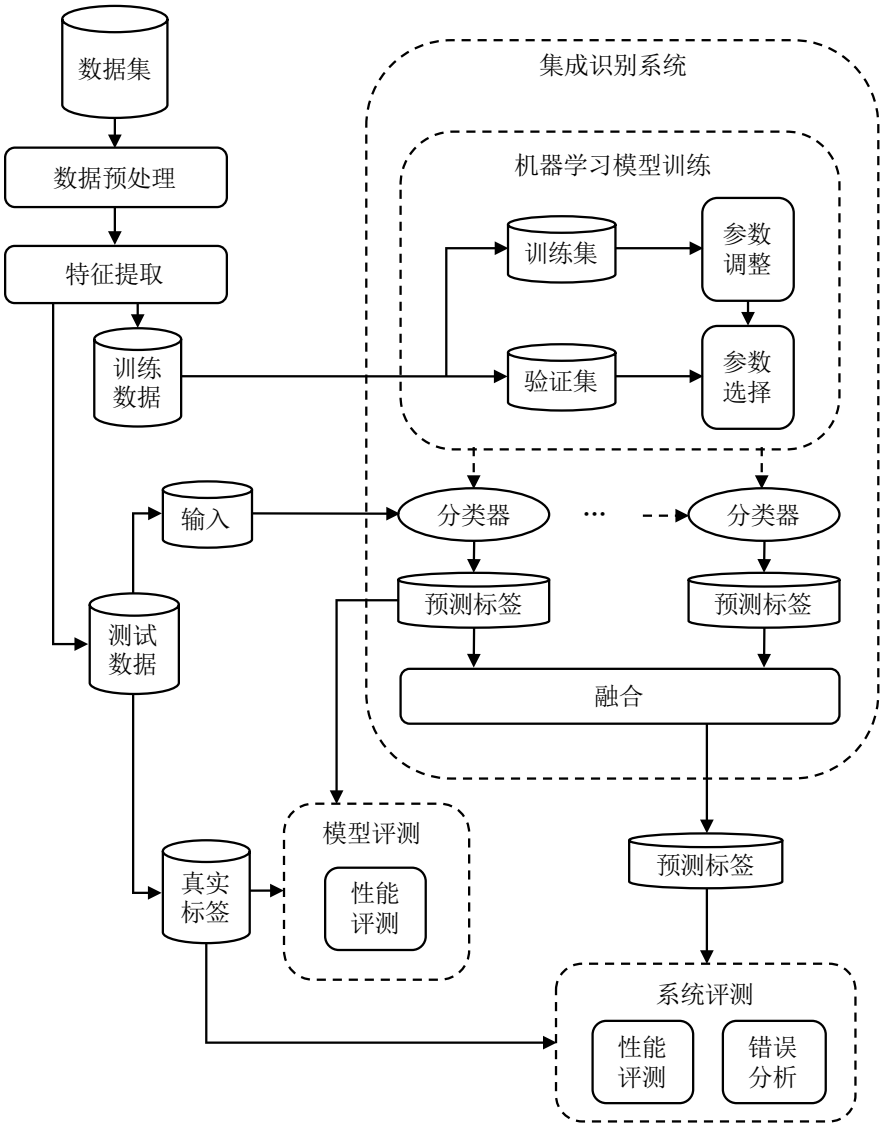


图 2.1　情感识别的研究框架

2.3　研究框架

在本小节，我们将给出一个情感识别分类问题的研究框架，并给出其中每一步需要完成的任务和目的。图 2.1显示本文面向情感识别的研究框架。

如图所示，整个框架的输入是实验的数据集，数据集的每个样本包含三个元素，分别是目标的文本、其对应的上下文以及原文本的情感类别标签，参考章节 2.2的形式化表示，每个样本可以表示为 $< t, b, c > \in T \times B \times C$。其中目标文本 t 和上下文 b 都需要首先进行预处理和特征提取以得出其量化后的特征，作为识别系统中各个模型的输入。然后整个数据集会被分成训练数据和测试数据，分别用于识别

系统的开发和评测。

对于一个多分类问题，我们可以把它分解成多个子分类问题的叠加。如面向正负中性情感的三分类问题，一些研究工作会首先判断文本的情感属于主观还是客观（中性），再将带主观情感的样本区分为正性和负性情感，即原本的三分类问题被拆解成主观和客观二分类以及正负性情感二分类两个问题的叠加。因此，我们再把接下来的研究工作分成两个阶段。

在第一阶段，我首先分别研究原问题所涉及的各个子问题，分析不同算法在各个子问题上的性能。这一阶段中对每个子问题的研究工作可以分成以下几步：

1. 基于不同模型得出分类器。而对于机器学习方法，标准流程是首先把训练数据分成训练集和验证集，利用训练集对模型的参数进行多轮的调整，每一轮分别得出一组模型的参数，然后评估每一组参数在验证集上的性能，最后取其中性能最好的一组参数作为分类器的参数。
2. 对测试集的样本进行预测。基于各个模型训练得出分类器后，分别预测测试集上各个样本所属的类别，得出预测标签。
3. 性能评测。对比预测标签和真实标签，计算各项评价指标。透过比较不同模型在各项指标上的差异，一方面可以了解各模型对该问题的建模能力，另一方面结合各模型的结构，从侧面了解语料背后的数学模型。

在第二阶段，我们将基于各个子分类问题组成一个集成识别系统，并分析其性能和特性。研究工作包括以下几步：

1. 为各个子分类问题训练分类器。基于第一阶段中对各个模型性能的分析结果，为各个子分类问题选定合适的模型，并训练一组分类器。
2. 对测试集的样本进行预测。对于测试集上的每一个样本，按照一定顺序由对应的分类器组回答各个子分类问题，以此得出最终的预测标签，同时保留其中的中间结果。
3. 性能评测。对比预测标签和真实标签，计算各项评价指标。另外还会观察系统的中间结果，了解每个子分类问题的设置对整个系统的识别能力的影响。
4. 错误分析。观察在测试集上样本被系统误判的原因，观察各个类别之间是否有明显的相互混淆的情况，透过深入分析指出系统的不足以及改进的方向。

在后续章节中的每组实验，我们都将基于此研究框架进行我们的工作。

2.4　相关技术

在本节中，我们将对研究框架中的文本预处理、特征提取、机器学习方法、集成学习方法进行介绍，分别说明它们的目的、具体技术和原理。对于在后续实验

用中使用到的技术，我们会给出相对充分的说明，同时也会为相关的技术给出概要的描述，以便读者在本论文未深入探索的方向进行拓展性的工作。

2.4.1　文本预处理

文本预处理是所有面向文本的研究的第一步，其目的是为特征提取做好准备。好的预处理策略可以在尽可能不掉失重要信息的情况下对样本数据进行简化，使样本之间重复的模式更容易被识别，降低算法对数据进行拟合的难度。错误的预处理策略会丢失具有区分能力的信息，甚至产生具有误导性的样本。对于不同的语言，由于存在语用和语法等方面的不同，预处理的方法也会各异。本小节将针对社交媒体上的英语文本介绍几种预处理技术。

2.4.1.1　分词

为了从文本提取词级别的特征，我们需要首先将句子分解成多个词的序列。

虽然对于英语及大部分西方语言，空格隔开的字符必然属于不同的单词，但对于不以空格分隔的语言（如中文），分词的作用尤其重要。在某些情况下，分词的结果会影响对句子的理解，如将“乒乓球拍卖完了”分解成“乒乓球拍-卖-完-了”或“乒乓球-拍卖-完-了”，对主体应该是“乒乓球拍”还是“乒乓球”，动词应该是“卖”还是“拍卖”，仅凭字面意思无法确定发言者想表达的意思。特别地社交媒体平台上，新词不断的出现，要正确进行分词就有其独特的难点。而分词并不只针对语言中的单词，还针对由标点符号和其他特别字符组成的有一定语义和情感的字符组合，如现今社交媒体上普遍用多个字符拼接成颜文字，其中最常见的有微笑的表情“:)”和伤心的表情“:(”，但如果在分词过程中把前者分割成“:”和“)”就会丢失其带有正向情感的信息，这在短文本的情感识别中非常关键。

虽然利用空格和标点符号在大部分情况下可以完成对英语句子的分词，但在一些情况下，标点符号作用为词组的一部分而不是分隔符，另外在社交媒体上也会有空格被省略的情况，以下是英语中一些需要额外处理的情况[40][41]。一是带句号的缩写，如“U.S.”、“.com”，句号应该作为词的一部分，如果按句号切割“U.S.”将失去其原本的语义。二是具有一定格式的带标点符号的词组，如电子邮箱地址（如 example@email.com）、时间（如 Jan 6th、06/01/19）、电话号码（如 (123)456-7890）、网页地址（如 www.example.com）等，这些内容在大部分情况下可以被视为一个个体，我们更关注这个个体对应什么类型的事物，而不是其细节，譬如将一个句子中电话号码的具体数字替换掉并不会改变其情感的表达，但识别出一个字符串对应的事情类型，并在清楚它对情感识别没有关联的情况下对其忽略是有意义的。第

三种情况是附属词，如“’t”对应“not”，只有正确识别“’”的作用才能识别出否定的意思，否则句子的意思将完全相反。值得注意的是，对社交媒体平台上的文本，除了以上在正规英语中会出现的情况，还会出现其他特殊情况，如“Y!E!S!”和“N!O!”。这需要对数据首先进行人工观察找出特殊的模式，再对语料库进行统计判断其出现频率，若出现频率较高则新增处理规则将对应模式做转换（如将“Y!E!S!”替换成“YES!!!”）或对问题无关的模式忽略（如国外某微博平台以“RE：”开头表示回复，并不包含任何情感，可以忽略）。

2.4.1.2　拼写修正

在处理较正式的文件时，我们一般可以默认其文本满足语言规范，词汇基本都是正确的拼写。不过在社交媒体上，拼写不符合规范的情况则非常普遍，这些情况可以分成三大类。

第一类是非刻意的拼意错误，由于社交媒体上的文本普遍是非正式的，用户不会刻意去保证文本的语言正确性，他们更关注于表达自己的想法和情感。拼写修正技术一般基于统计的方法，对于一个不存在的词汇，尝试对其拼写进行有限次编辑，再评估编辑后的词在其上下文中出现的概率，最后选出可能性最高的一个词作替换。拼写修正技术已相当成熟地被应用于搜索引擎和手机键盘等应用中，读者可参考相关研究[42] [43] 了解其细节。

第二种情况是网络上常用的代替用词，如以“thx”、“tnx”代替“thanks”，以“k”代替“ok”，以“cant”代替“can’t”。虽然这些用法没有被认可为标准用法，但由于方便而在网络上被广泛传播，成为网民们都能理解和接受的用法。对此的处理方法有两种，一是人工建立映射表，把代替用词映射到标准的英语词汇，优点在于可以直接引用标准词汇的语义信息，但缺点是需要人工参与，特别是在网络上新用词不断出现的情况下需要持续的更新。另一种是不做处理，利用机器学习方法从大型语料库中自动学习出它的语义，优点在于省去了人工的部分，缺点在于新词的出现很稀疏，严重依赖于语料的收集。

第三种情况是语气加强，如“yeeeees”、“AMAZING”，但对于这一类情况，处理的重点不只在于转换成标准用词，而是根据具体的研究问题判断是否要识别这种语气的加强作为提示信息，譬如在情感识别中，语气加强一般提示了此处表达的情感较强烈。Baziotis 等人[44] 的做法是在单词前后添加标签示意，如“yeeeees”替换成“yes <enlongated>”，“AMAZING”替换成“amazing <allcaps>”。

2.4.1.3　规范化

在章节2.4.1.1中曾提及在一些特定的应用场景下，电子邮箱地址和日期等内容的细节其实对我们的研究任务并不起作用，因此我们可以用一个特定的单词代替相同类型的内容。如 Kouloumpis 等人[45] 在处理微博文本时将不同的网页地址和“@ 用户名”分别用特定的字符串代替。更进一步地，Baziotis 等人[44] 在研究情感识别时，将表达同一种情感的不同表情符替换成对应的情感标签，如“:)))”替换成“<happy>”, “:-D”替换成“<laugh>”。但需要注意替换的内容是否会导致信息掉失，如 Joshi 等人[46] 在针对反讽识别的研究中指出一些反讽透过具体数字的对比来表达，在这种情况下将具体数字统一替换成表示数字的标签可能会导致预处理后的样本失去原本的反讽属性。

2.4.2　特征提取

数据经过预处理后，我们需要从获得的字符串或字符串序列中提取任务相关的特征，并进行量化方可作为分类器的数学模型的输入。本小节将介绍几种主流的文本特征提取技术。

2.4.2.1　词嵌入

词嵌入即捕足词的语义、语法、情感等信息并将其以向量表示。其中词对应预处理中分词得到的词，或对于字符级别的模型，“词”可以对应单个字符[47]。主流的词嵌入算法分成两大类：基于人工神经网络的方法和基于矩阵分解的方法。

基于人工神经网络的方法假设句子中词之间存在某种上下文关系，并以神经网络建模学习，其中最具代表性的是 CBOW 和 Skip-gram 两种算法。CBOW 的建模原理是句子中的一个“词”可以由它在句子中的前后有限个“词”决定，Skip-gram 的建模原理则是句子中的“词”可以推测出它在句子中的前后有限个“词”。这两种算法都在自然语言处理中常用的词嵌入学习工具 Word2vec[48] 中提供，又由于训练优质的词嵌入模型（指一组词到词向量的映射）需要收集大量的数据和耗费相当的时间，一些利用 Word2vec 训练好的词嵌入模型会作为公开的资源供其他研究员使用。其中较常用的是谷歌提供的模型①，利用谷歌新闻作为语料训练出复盖约三百万个词的三百维向量集合。然而词向量所包含的信息和语料库紧密相关，所以也有一些研究会按照自己课题的特性自行收集相应的语料来训练词向量。如 Baziotis 等人[44] 利用微博平台 Twitter 上收集的数据训练词向量模型并应用于面向微博的

① https://code.google.com/archive/p/word2vec/

反讽识别，其词向量模型也被公开供其他研究者使用①。

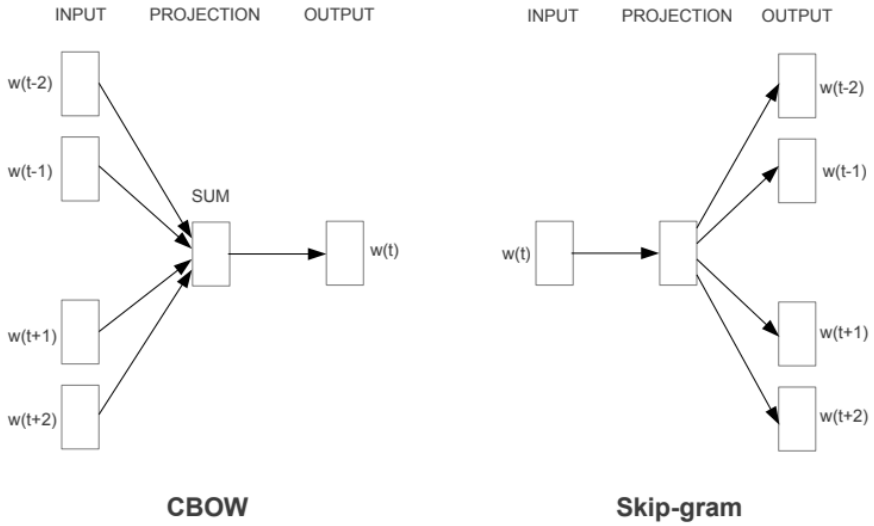


图 2.2　Word2vec 提供的两个算法的模型，引自 [48]

基于矩阵分解的方法假设一个表示词和词或词和文档之间关系的矩阵可以表示为由词向量组成的矩阵和其他矩阵的相乘，因此对该关系矩阵以对应方式进行矩阵分解即可以得到词向量。具代表性的方法是隐藏语义分析方法（Latent semantic analysis，LSA）[49]，以词在各个文档中出现的频率得出词-文档矩阵，并假设其文档可以表示为词-词概念、词概念-文档概念、文档概念-文档三个矩阵相乘，利用奇异值分解（Singular Value Decomposition，SVD）得到词-词概念矩阵，取每一行作为对应词的词向量。另外还有 Pennington 等人 [50] 提出的 GloVe 算法，该算法根据不同词在语料中出现在各自上下文窗口中的频率得出词和词之间的关系矩阵，再分解成两个词-词概念矩阵的相乘。其研究结果显示在部分自然语言处理任务中，采用 GloVe 较 Word2vec 及另外几种词嵌入技术得出的词向量能达到更好的效果。他们同样公开了多个利用不同语料库训练好的词向量模型供其他研究者使用②。

2.4.2.2　词汇特征

词汇特征表示文本中一元或多元语法的分布情况，不考虑词的具体意思。譬如常用的词袋模型（Bag of words, BOW），其将一段文本映射到一个固定长度的向量，向量上每一维对应一个一元或多元语法的某种属性，譬如它是否在文本中出现（即独热编码），或它在文本中出现的频率，或其词频-逆文档频度 (Term Frequency

① https://github.com/cbaziotis/ntua-slp-semeval2018
② https://nlp.stanford.edu/projects/glove/

- Inverse Document Frequency，TF-IDF)。或更简地以单词数量或字符数量作为一维的特征。

2.4.2.3　句法特征

句法特征表示文本中各个词在句子的语法作用。常见做法是先对文本进行词性（Part-of-speech，POS）标注和依赖树分析，再基于标注提取特征，如形容词的数量、副词的数量，或以标注代替单词后采用词袋模型得出固定长度的特征向量。一些研究会针对性地手工设计句法特征，这要求研究者对语法有所了解并挖掘它对研究问题是否存在关联性，相对地这种特征的解释性更强。

2.4.2.4　语义特征

语义特征基于文本的字面意思，表示文本表达的内容或其中的个别属性 （如主题分布），这更接近于人们透过理解句子内容来进行识别。常见技术譬如基于文本中各个词所对应词向量计算句子的表示向量，或利用词向量计算句子中各单词和情感词的相似度来间接评估词的情感极性，或利用聚类算法对语料库的单词进行聚类来获得单词的隐藏语义类别，再计算句子中各类别单词的分布。

2.4.3　机器学习方法

本小节将给出搭建情感识别系统相关的机器学习方法。目前用于情感识别的主流算法可以分成两大类：传统机器学习和人工神经网络。我们会分别对这两类算法中具代表性的例子给出介绍。

2.4.3.1　传统机器学习

支持向量机（Support Vector Machine, SVM)[51] 由 Cortes 和 Vapnik 在 1995 年提出，是目前最常用的传统机器学习算法之一。在对二分类问题的建模过程中，每个训练样本被映射到一个空间中的一个点，而支持向量机尝试找出一个边缘把空间分成两个子空间（如以直线分割二维空间，以平面分割三维空间），每一个子空间对应一个类，使得训练样本尽可能落在对应类别的子空间中。对于多分类问题，主流做法是把原问题拆解成多个二分类问题。后随着核函数的引入，将支持向量机拓展成非线性的数学模型，大大加强了其建模能力。除了分类问题，支持向量机同样可以应用于回归问题，Drucker 等人[52] 则提出了支持向量机的回归模型。支持向量机最大的限制之一是只接受固定长度的向量作为输入，要求从输入数据提

取出固定长度的特征向量，换言之基于支持向量机的识别系统非常依赖对特征的提取。

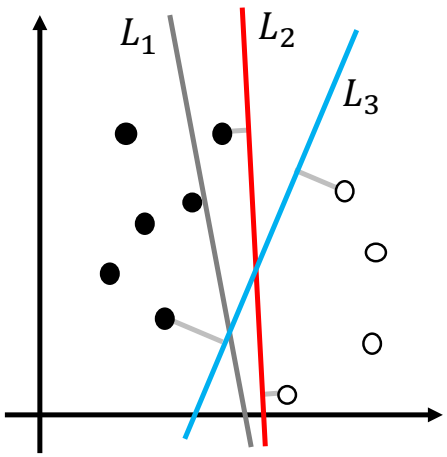


图 2.3　支持向量机在处理二类问题的示情感。其中黑点和白点分别对应两个类别的样本，L_1 不能对两类样本作区分，L_2 和 L_3 均成功区分两个类别的样本，但 L_3 和两个类别样本的距离更大，为更优解

决策树（Decision Tree）是另一种常见的机器学习方法。对于分类问题，决策树基于训练数据建立一个树状模型，其中每个非子叶结点对应一个决策，叶子结点对应要识别的类别。在识别阶段，识别过程从根结点开始，经过每个非叶子结点根据输入数据得出决策结果，不同的决策结果会对应下一级的一个结点，如此不断前进直到到达叶子结点，以其对应类别作为分类结果。和支持向量机一样，决策树的输入为固定长度的向量，要求对应的特征提取。现今的研究更多地会使用由多个决策树组成的随机森林（Random Forest）而非单个决策树作为分类器，随机森林属于集成学习一类，将在后续章节2.4.4中介绍。

2.4.3.2　人工神经网络

人工神经网络源自仿生学，以数学模型模拟生物神经网络的响应过程和学习过程。Rosenblatt 提出的感知器[53] 是最早的人工神经网络，也是现在人工神经网络的基本单元（又称为神经元）。Rumelhart 等人[54] 在后来提出了一种多层前馈神经网络，BP 网络，由多层的感知器和激励函数反复堆叠而成。更重要的是他们提出了它的学习算法，反向传播算法（Back-propagation，BP），当网络的输出结果和预期结果不同，透过构造损失函数计算偏差，再根据损失函数和偏差计算出神经元

权重的梯度，其对应了使误差扩大的权重方向，以其反方向作为权重修改的依据。同时将该层输入的修改量作为前一层输出的偏差，逐层重复修改权重，整个过程重复迭代直到误差满足一定条件时学习过程结束。反向传播算法是目前人工神经网络模型训练中的核心算法，但随着网络层数增加，梯度以指数级别增大或缩小，会导致梯度大小过大或接近于零的情况，即梯度爆炸和梯度消失。直到 Hinton 等人[55] 提出深度信念网络和无监督逐层训练的策略，其核心思想是在每一轮训练只调整网络中一层的权重，透过固定前一层的输出作为本层的输入，避开了多次传播带来梯度的指数级变化。如此网络逐层经过调整后，再对整个网络进行权重调重，理论上此时的梯度较小，避免了梯度爆炸的出现。这使得现在复杂人工神经网络的学习成为了可能，深度学习正式进入高速发展的阶段。接下来我们将对主流的人工神经网络单元作出介绍。

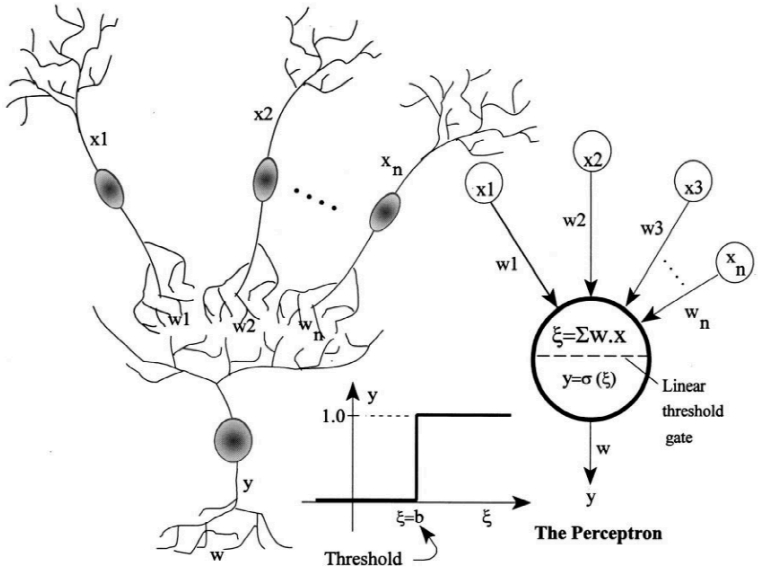
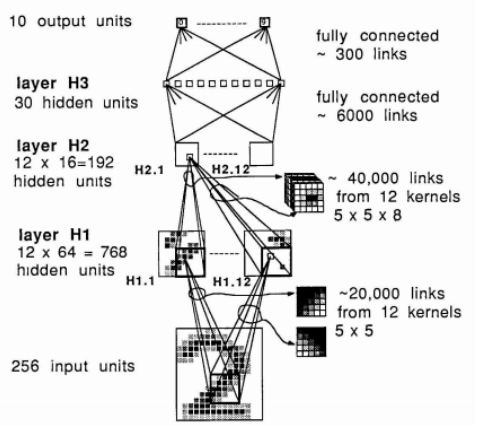


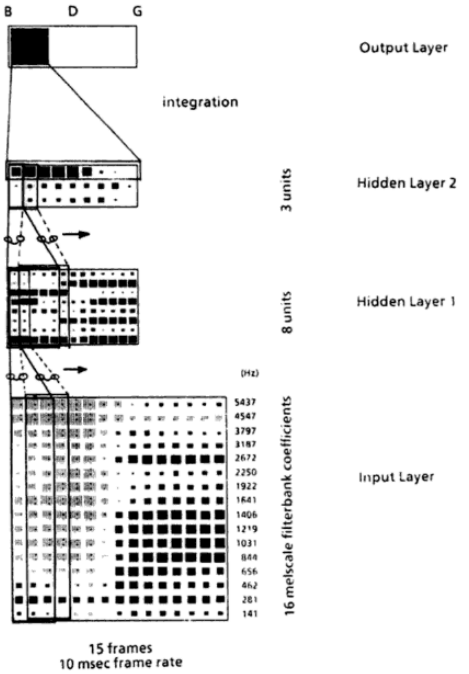
图 2.4　生物的神经元与感知器的模型示意图，引自[56]

卷积神经网络（Convolutional Neural Network，CNN）是目前最常用于计算机视觉的一类人工神经网络，它的设计来源自生物的视觉皮层。Hubel 和 Wiesel[59] [60] 在对猫和猴子的视觉皮层进行观察时发现其中一些神经元只对接收到的视觉画面的一个区域作出反应，该区域称为该神经元的感受域（Receptive Field）而相邻神经元的感受域会有所重叠，联合构成对整个视觉画面的接收和反应。这些神经元中再细分为两类，第一类神经元对画面中特定方向的刺激反应最强烈，另二类神经元则较前一类神经元的感受域大，但其反应对刺激在感受域中的具体位置不敏感。

基于以上生物特性，Fukushima[61] 提出了神经认知机（Neocognitron），其数学



(a) Lecun 等人[57] 的反向传播网络



(b) Waibel 等人[58] 的时延神经网络

图 2.5　卷积神经网络模型

模型中包含了两个关键结构，卷积层和降采样层，与前面两类神经元的功能一一对应。受启发于神经认知机，Lecun 等人[57] 提出了一个三层的反向传播网络，包含特征映射和降采样两种功能的元件，同样与视觉皮层中的两类神经元对应。该网络被应用于手写数字的识别，和当时同类型网络的性能相比达到了近 30% 的提升，该数学模型自此成为了现今卷积神经网络的原型。

虽然卷积神经网络起源于视觉处理，但其变型还应用于其他领域。如 Waibel 等人[58] 提出的时延神经网络（Time-Delay Neural Network，TDNN）被应用于计算机听觉的音素识别，其核心思想是将每一个时刻的音频特征按时间顺序堆叠，以时间上的相邻类比视觉画面上位置的相邻。Zhang 等人[62] 提出了一个字符级别的卷积神经网络并应用于文本主题分类，以字符类比语音中的信号，首次将卷积神经网络用于自然语言领域。

递归神经网络（Recurrent Neural Network，RNN）是另一个人工神经网络的大类，和前馈神经网络最大的区别在于递归神经网络中以某种机制实现了记忆能力来保留前一时刻的状态，因此可以对一个序列输入中前后出现的内容进行关联，换言之对序列的数据进行建模。递归神经网络会在每一时刻接受一个输入，并结合其“记忆”计算这一时刻的输出，故递归神经网络的输出是和输入序列长度相同的序列。递归神经网络的应用场景包括视频内容识别、文本翻译、股票指标预测等。较早期被提出的递归神经网络模型是 Elman[63] 的 Elman 网络，其数学模型如下：

$$h_t = \sigma_h(W_h x_t + U_h h_{t-1} + b_h) \tag{2-1}$$

$$y_t = \sigma_y(Y_y h_t + b_y) \tag{2-2}$$

其中 $x_t \in \mathbb{R}^d$ 为 t 时刻的输入，$h_t, h_{t-1} \in \mathbb{R}^h$ 分别表示网络在这一时刻和上一时刻的隐藏状态，$y_t \in \mathbb{R}^y$ 为网络的输出，σ_h 和 σ_y 为激活函数，$W_h \in \mathbb{R}^{h\times d}, U_h \in \mathbb{R}^{h\times h}, b_h \in \mathbb{R}^h, Y_h \in \mathbb{R}^{y\times h}, b_y \in \mathbb{R}^y$ 为模型参数。可见在模型中因为引入 h_{t-1} 作为当前时刻的输入而表现出某种记忆，上一时刻的输入 x_{t-1} 间接影响了这一轮的输出。

目前常用的递归神经网络之一是 Hochreiter 和 Schmidhuber[64] 提出的长短时记忆模型（Long Short-Term Memory，LSTM），其数学模型如下：

$$f_t = \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \tag{2-3}$$

$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \tag{2-4}$$

$$o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \tag{2-5}$$

$$c_t = f_t \circ c_{t-1} + i_t \circ \sigma_c(W_c x_t + U_c h_{t-1} + b_c) \tag{2-6}$$

$$h_t = o_t \circ \sigma_h(c_t) \tag{2-7}$$

其中 $x_t \in \mathbb{R}^d$ 为 t 时刻的输入，h_t，$h_{t-1} \in \mathbb{R}^h$ 分别是这一时刻和上一时刻 LSTM 的输出（或当 LSTM 作为一个人工神经网络中其中一层时称为隐藏状态），$c_t \in \mathbb{R}^h$ 为当前时刻记忆，i_t，f_t，$o_t \in \mathbb{R}^h$ 分别控制了这一时刻的输入、上一时刻记忆和这一时刻的输出在这一刻起的作用，因此称为“输入门”、“遗忘门”、“输出门”的激活向量，$\sigma_g, \sigma_c, \sigma_h$ 为激活函数，特别地 σ_g 一般采用 $sigmoid$ 函数，σ_c, σ_h 一般采用 $tanh$ 函数，$W \in \mathbb{R}^{h\times d}, U \in \mathbb{R}^{h\times h}, b \in \mathbb{R}^h$ 为网络参数。和 Elman 网络比较可见，LSTM 除了有 h_{t-1} 传递上一时刻的信息以外多了一个记忆单元 c，具有更强的记忆功能。在 LSTM 的设计中包含了“输入门”、“遗忘门”、“输出门”和记忆单元，对序列数据有较强拟合能力，相对地对于小规模的序列数据容易出现过拟合。

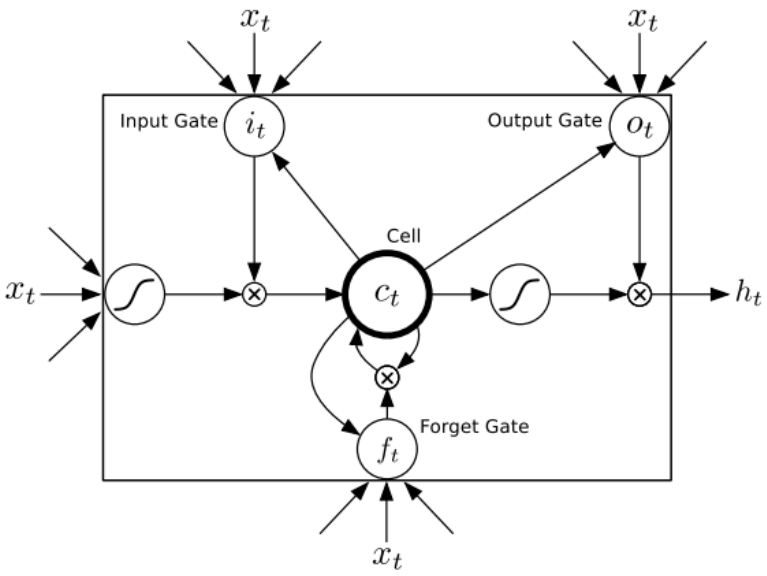


图 2.6　长短时记忆模型，引自[65]

为了解决容现出现过拟合的问题，Cho 等人[66] 提出了门控循环神经元（Gated Recurrent Unit, GRU）。可以认为它是 LSTM 的一种轻量级变型，其数学模型如下：

$$z_t = \sigma_g(W_z x_t + U_z h_{t-1} + b_z) \tag{2-8}$$

$$r_t = \sigma_g(W_r x_t + U_r h_{t-1} + g_r) \tag{2-9}$$

$$h_t = (1 - z_t) \circ h_{t-1} + z_t \circ \sigma_h(W_h x_t + U_h(r_t \circ h_{t-1}) + b_h) \tag{2-10}$$

可见在数学模型上 GRU 和 LSTM 非常相似，虽然 GRU 中省去了专门的记忆单元，但同样采用了“门”来控制当前时刻的输入和前一时刻输出的作用。GRU 的模型结构要较 LSTM 的简单，而在同一组超参数下，GRU 的模型参数要明显少于 LSTM，故理论上在处理较小规模的序列数据时更有优势。

从上面的数学模型中我们不难发现，时刻 t 的输出只受时刻 t 及以前的输入影响，不过在一些场景下我们会考虑时刻 t 以后的输入是否也会与时刻 t 的输出有关。譬如在文本情感识别时，英语有一种后置定语的语法，在句子中一个单词 A 由出现在它后面的一组单词来修饰，那么为了正确理解在时刻 t 出现的单词 A，我们就需要引入时刻 t 以后的信息。为此 Schuster 和 Paliwal[67] 提出了双向递归神经网络（Bidirectional Recurrent Neural Network，BRNN），如图2.7所示，一个双向递归神经网络包含两个 RNN，对于一组序列数据，分别以其原本方向和反方向输入到两个 RNN 中，从而获得两个方向上与 t 时刻相关的信息。根据此思想，RNN 可

以替换成任何具体的递归神经网络如 LSTM 和 GRU。

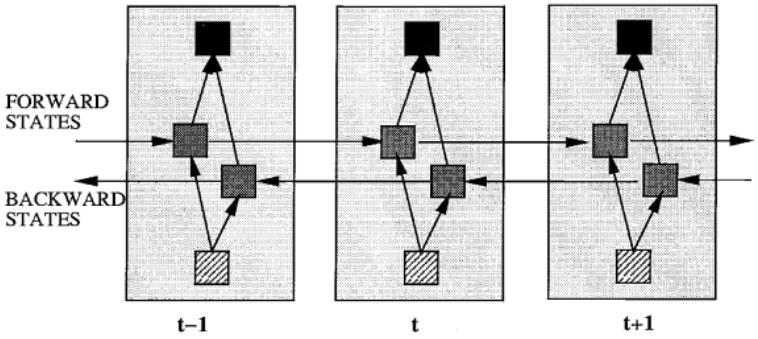


图 2.7　双向递归神经网络，引自 [67]

注意力机制（Attention Mechanism）受启发于人类的视觉处理机制，当人们尝试理解一个画面的内容时，他们不会马上对画面里的所有细节都进行处理，而是首先找出相关信息可能在的部位，再对该部分的内容作进一步处理。当人们对同类型画面有一定观察后，会总结出关键信息出现的模式，并在处理新的画面时优先采用这些模式来理解其中的内容。Mnih 等人 [68] 以此提出了结合递归神经网络和注意力机制来实现对手写数字的识别，不是透过预处理而是直接依赖注意力机制找出图片中数字所在的位置。其后 Bahdanau 等人 [69] 同样尝试了在递归神经网络中加入注意力机制，但这次的应用场景从计算机视觉转移到了自然语言处理的机器翻译，其注意力机制的模型如下：

$$e_{ij} = a(s_{i-1}, h_j) \tag{2-11}$$

$$\alpha_{ij} = \frac{exp(e_{ij})}{\sum_{k=1}^{T_x} exp(e_{ik})} \tag{2-12}$$

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j \tag{2-13}$$

其中 s_{i-1} 表示上一时刻（$i-1$）的隐藏状态，h_j 表示在时刻 j 的输入信息，a 是一个计算 s_{i-1} 和 h_j 关联性的函数，$\alpha_{ij} \in (0, 1]$ 表示对输出时刻 i，给予输入时刻 j 的信息的注意力，最后根据各个时刻应该给予的注意力结合所有输入信息，得出用于时刻 i 输出的相关信息 c_i。其后 Vaswani 等人 [70] 针对机器翻译提出了一种仅基于注意力机制搭建的人工神经网络，实验结果显示他们的模型超越了该实验数据上曾达到过最好的性能。虽然在近年不同研究 [71] [72] [73] 中同现了各种注意力机

制的变型，但这些模型可以统一描述如下：

$$\alpha_{ij} = a(q_i, k_j) \tag{2-14}$$

$$c_i = \frac{\sum_{j=1}^{T_x} \alpha_{ij} v_j}{\sum_{j=1}^{T_x} \alpha_{ij}} \tag{2-15}$$

其中 q_i 表示输出时刻 i 对应的一个请求，k_j 和 v_j 表示输入时刻 j 的一个键值对，a 是计算请求和键之间关联性的函数，以得出对于输出时刻 i 应给予输入时刻 j 的信息 v_j 的注意力 α_{ij}，最后以加权平均计算出输出时刻 i 相关的信息 c_i。一些研究[20][21] 则针对其中的隐藏状态 α_{ij} 分析注意力机制的具体运行，并得出了和人工理解一致的结果。

2.4.4　集成学习

集成学习（Ensemble Learning）的目的在于结合多个模型以达到比其中任何一个模型都要好的识别性能。集成学习方法包含了多个类别，其中常用于分类问题的是基于投票的方法，譬如多数投票（Majority Voting）、加权多数投票（Weighted Majority Vote）、加权平均概率投票（Soft Voting）。假设对于 N 分类问题，有 T 个基础模型 $h^i, i = 1, 2, ..., T$。对于输入数据 x，基础模型 h^i 的输出如下：

$$h_j^i(x) = \begin{cases} 1, & \text{若模型} h^i \text{将} x \text{识别为类别} j \\ 0, & \text{其他} \end{cases} \tag{2-16}$$

对于多数投票，其最终预测结果 $H_M(x)$ 如下：

$$H_M(x) = \arg\min_j \sum_{i=1}^{T} h_j^i(x) \tag{2-17}$$

在章节 2.4.3.1中提到的随机森林就是以决策树为基础模型，再采用多数投票得出最终识别结果的算法。对于加权多数投票，其最终预测结果 $H_{WM}(x)$ 如下：

$$H_{WM}(x) = \arg\min_{j} \sum_{i=1}^{T} w_i h_j^i(x) \tag{2-18}$$

其中 $\sum_{i=1}^{T} w_i = 1, w_i \geq 0$ 为每个模型的权重，相当于假设不同模型具有不同的可信度。而加权平均概率投票要求模型能给出样本在不同类别上的概率分布 $p^i(x) \in [0,1]^N$，其最终预测结果 $H_S(x)$ 如下：

$$H_S(x) = \arg\min_{j} \sum_{i=1}^{T} w_i p_j^i(x) \tag{2-19}$$

其中 $\sum_{i=1}^{T} w_i = 1, w_i \geq 0$，和加权多数投票中 w_i 的原理相同。

2.5　本章小结

在本章中，我们首先给出了面向文本的情感识别的形式化表示，描述了其中所涉及的要素并给出了对应的例子，在本论文的后续章节中，我们将利用此形式化表示套用到我们要解决的各个具体问题上，以显示各个问题之间的共性。然后我们提出了一个面向文本情感分类问题的研究框架，该研究框架的核心思想是把一个多分类问题拆解成多个子分类问题的叠加，首先对各个子分类问题进行研究，再结合各个子分类问题来解决原本的多分问题。最后我们对研究框架中涉及到的一些技术领域进行了说明，介绍了这些技术领域的目的、原理以及目前的一些主流技术，有助于读者更好的理解后续的实验以及进行拓展性的工作。

第 3 章　基于多步决策的微博反讽识别

3.1　本章引论

社交媒体的发展对我们的语言体系带来了很大的影响，网络上不断出现新的用词和句式，语言的表达方式越来越丰富，同时变得越来越复杂，这也是面向社交媒体的文本情感识别的难点之一。而其中，反讽是在网络上常见的语言修辞手法之一，它在情感识别的研究当中有着重要的地位。Henry Watson Fowler 在《The King's English》一书中指出反讽的使用使得"表面意思和实际意思不同"。譬如一个人说"你这想法真有创意"，在字面意思上是对另一个人的赞同，但在特定背景下，如果紧接一句"你真相信这能实现吗"，那么发言者实际上可能暗示这个想法无法落地，表面上称赞为"有创意"，其实在暗示这种想法不切实际。这在情感识别当中尤其重要，无视反讽的使用会导致对内容的错误理解，因此反讽识别一直是和情感识别紧密相关的研究课题之一。

根据 Joshi 等人[74] 对近年相关研究的总结，反讽识别可以大致分成基于规则的方法和基于机器学习的方法两种。基于规则的方法透过人工找出反讽修辞的语言规律，设计出对应的匹配规则，然后在识别过程中，若文本中的内容符合匹配规则，则认为它属于对应的反讽类别。和机器学习方法对比，基于规则的方法优点在于无需模型训练，但要求研究员在语言知识上对反讽有充分的理解，设计的匹配规则对语言模式的复盖程度决定了算法的识别能力。而随着近年深度学习快速发展，一些研究更关注词嵌入向量的使用以及人工神经网络的设计。

在反讽识别的研究中，各类别样本分布不均匀是一个备受关注的问题。虽然反讽是较常见的修辞手法，但在现实场景中，使用反讽的情况还是占少数。对于反讽的细分类别，其样本分布不均匀的情况则更为明显。但倾向于把样本识别为占比较大的类别是目前机器学习算法的通病，这也是其他研究领域在现实应用场景中会出现的问题。针对数据不均匀的情况，主流的解决方法包括欠采样、过采样、数据增强等。虽然这些方法在语音和图像领域使用较广，原理较直观而实现上也较简单，但在自然语言处理领域，因为要考虑文本的语义和语法，目前还没有较成熟的做法。为此我们提出了一种基于多步决策的多分类系统框架，有别于从数据的采样或对数据的加工入手，我们把多分类问题拆解成多个子分类问题的叠加，再经由回答每个子分类问题来得出原本多分类问题的识别结果。透过拆解出子分类问题，使得在局部调整识别系统的性能成为可能。

国际比赛 SemEval-2018 的任务三[36] 旨在促进英语微博中的反讽识别研究，其

中包含了两个子任务。子任务一是二分类的反讽识别，需要识别微博是否有使用反讽。子任务二是四分类的反讽识别，是子任务一的拓展，除了判断微博是否包含反讽，原本的反讽类别再细分成三个类别：基于相反语义的反讽、情景反讽、其他反讽。本章节中我们将基于 SemEval-2018 的任务三进行实验，并透过和其他参赛系统进行比较来评估我们的系统性能。

本章的内容安排如下。在章节3.2中，我们会首先给出当前问题的形式化表示。在章节3.3中我们再对具体实验数据进行观察，分析各个反讽类别样本的特征以及微博文本的特征等。章节3.4将给出我们针对当前问题提出的基于多步决策的识别系统框架，以及组成该系统的模型结构。最后在章节3.5，我们会给出实验的细节以及对实验结果的分析。

3.2　形式化表示

在本章中，我们要研究面向微博的反讽类型识别，以下我们对应章节 2.2给出此问题的形式化表示。给定一个反讽类别集合 C，对于微博集合 T 中的任意一条微博 t，它属于唯一一种反讽类别 $c \in C$。又给定一个词集合 W，微博 t 经过文本预处理后可以表示为一个长度为 L 的词序列 $w =< w_1, w_2, ..., w_L >, w_i \in W, i \in [1, L]$。因为没有引入上下文信息，所以背景 B 在模型中忽略。那么我们的目标就是找出一个映射关系 F_C，使得 $c = F_C(w)$。

3.3　实验数据

我们的实验采用 SemEval-2018 的任务三提供的数据集，其中的语料收集自微博平台 Twitter 上发布于 2014 年至 2015 年之间的微博，再由人工标注得出每条微博的反讽类型。该比赛的两个子任务均采用了相同的语料，但标注稍有不同。

3.3.1　样本分布

子任务一是二分类的反讽识别，需要识别微博是否有使用反讽，各类别的样本分布如表3.1所示，表3.2为语料中两个类别的例子。可以看出“没有反讽”和“带有反讽”两个类别的样本在训练集上大致比例为 1:1，在测试集上两个类别的分布大致为 3:2。

子任务二是四分类的反讽识别，是子任务一的拓展，除了判断微博是否包含反讽，原本“带有反讽”的样本再细分成三个类别：基于相反语义的反讽（为简化，在以下图表中简称为“反义反讽”）、情景反讽、其他反讽。各类别的数据分布如

表 3.1　反讽识别子任务一各类别样本数量分布

数据集	没有反讽	带有反讽
训练集	1923	1911
测试集	473	311

表 3.2　反讽识别子任务一样例

编号	类别	例子
1	没有反讽	Had no sleep and have got school now #not happy
2	带有反讽	I just love when you test my patience!! #not

表3.3所示，表3.4为语料中各反讽类别的例子。可见“带有反讽”一类细分出的三个反讽类别在样本分布上有明显的不均匀，“基于相反语义的言语反讽”占了其中一半以上，“情景反讽”、“其他反讽”的样本量则相对较少。

表 3.3　反讽识别子任务二各类别样本数量分布

数据集	没有反讽	反义反讽	情景反讽	其他反讽
训练集	1923	1390	316	205
测试集	473	164	85	62

3.3.2　各反讽类别的语言特征

比赛组织者对四种反讽类别给出了对应的说明。对“基于相反语义的反讽”，文本中存在某部分内容表达了可评估的情感极性，但整条微博实际上表达了相反的情感极性。如表3.4中的例子 2，“love”在字面意思上表达了正面的情感，但微博后半中“awful weather”提示实际情况引起了发言者的不适，发言者其实在表达对这个夏天坏天气的不满，这和“love”的正面情感恰恰相反。对于“情景反讽”，文本正描述某个场景，其中发生的事情和对该场景的预期不符。如表3.4中的例子 3，描述了一个参与讲座的场景，但“我们”并没有专注于这场讲座，和“参与者应该专注于讲座内容”的预期相反。对于“其他反讽”，文本表达了讽刺的意思，但文本的字面意思和发言者表达的意思之间并不存在情感极性的反差。如表3.4中的例子 4，发言者表示某人想要保持蟋蟀干净，字面上并不存在情感极性，但在句子后的井号标签提示了发言者表达了讽刺，认为“保持蟋蟀干净”是一样莫名奇妙的事情。最后是“没有反讽”一类，对于明显不可能包含反讽的文本，或者在背景信息不足的情况下不能确认其包含反讽的文本均属于这一类。

表 3.4　反讽识别子任务二样例

编号	类别	微博
1	没有反讽	Had no sleep and have got school now #not happy
2	反义反讽	I really love this year's summer; weeks and weeks of awful weather
3	情景反讽	Most of us didn't focus in the #ADHD lecture. #irony
4	其他反讽	@someuser Yeah keeping cricket clean, that's what he wants #Sarcasm

3.3.3　文本长度

我们对数据集的文本进行分词后统计了各类别样本的单词数量分布，以下简称为文本长度。表3.1和表3.2分别显示了训练集和测试集上各类别样本的文本长度。综合先见样本的文本长度不超过 50，样本的平均文本长度约为 20 个词。根据表3.1我们可以看出“情景反讽”的样本整体的文本长度较其他类别的长，“没有反讽”和“基于相反语义的反讽”在文本长度分布上没有明显区别，“其他反讽”整体的文本长度则略高于前两者。再观察表3.2，同样地“没有反讽”和“基于相反语义的反讽”在文本长度分布上没有明显区别，“情景反讽”的文本长度略长于前两者，但不如训练集上明显。

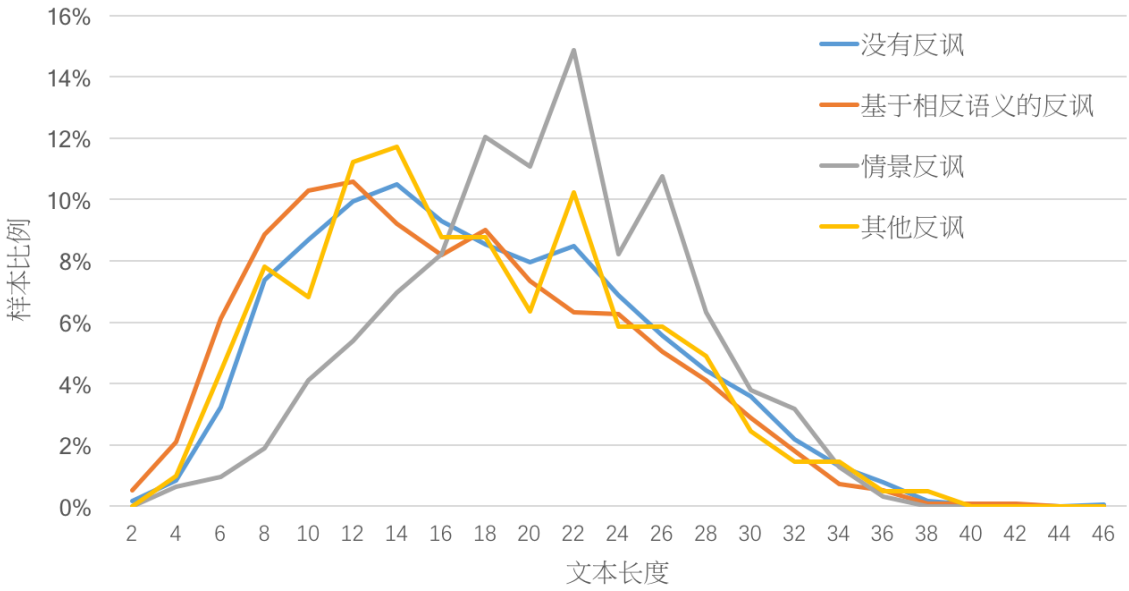


图 3.1　反讽识别训练集上各类别文本长度分布

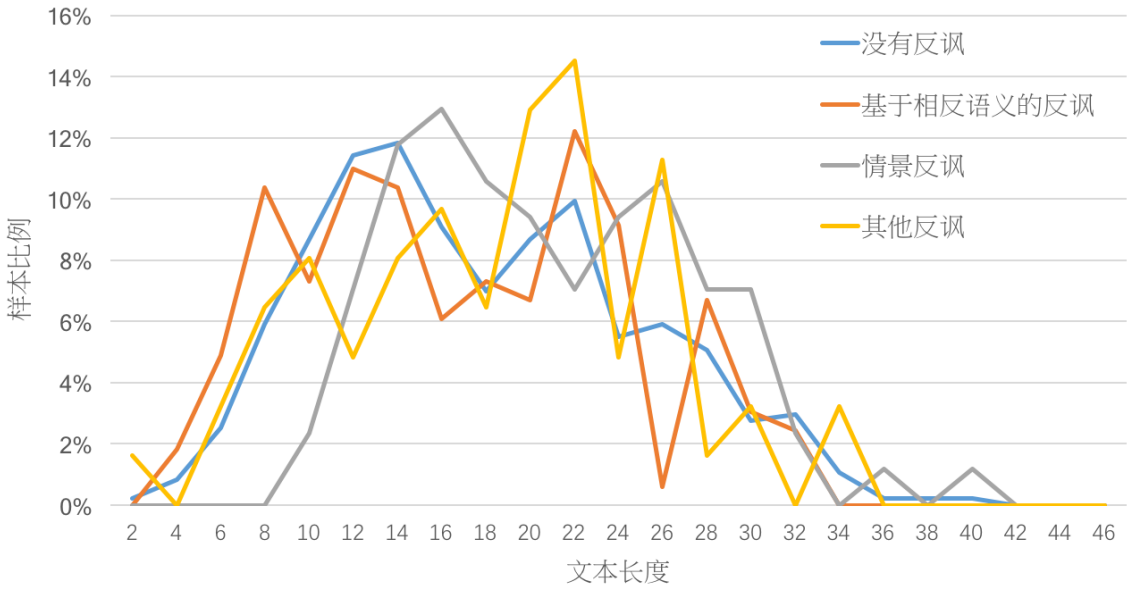


图 3.2　反讽识别测试集上各类别文本长度分布

3.3.4　文本特征

另外我们注意到语料中存在微博平台 Twitter 上特有的文本特征，出现频率较高的特征如下：

- 用户标签“@someuser”。对应微博上的一个用户，使用场景包括以下两种。一是作为句子中的名词使用，二是添加在句前或句末，用于提示该用户与本条微博有关，不具有语法作用。
- 井号标签“#something”。使用场景大致分为以下两种。一是作为句子中的一部分，如“I #do #like #it”，去除井号后满足正规的英语用法，此时井号标签的作用是对内容的强调。另一种是出现在句末，用于提示微博内容与标签对应内容相关，如句末出现“#sarcasmtweet”则是明示讽刺。
- 网站链接。在微博平台上支持附加一个网站链接，其中平台对链接进行了统一处理，在语料库中网站链接均形如“http://t.co/***”或“https://t.co/***”。
- 转发标记“RT”（retweet）。当用户转发一条微博并添加个人评论时，平台会自动在个人评论后附加转发的微博原文并以“RT”隔开。
- 一些在社交媒体平台上常见的、有别于正规英语的用法，如拼写错误、缩略词、全大写字母的单词、表情符等，可以参考章节2.4.1描述的例子。

3.4　框架设计

对于任务一，由于是二分类问题，我们只考虑使用一组二分类器进行多数投票得出最终的预测结果，判断微博文本是否采用了反讽修辞。

对于任务二，我们提出了一个基于多步决策的识别系统。对于原本的反讽四分类问题，我们把它拆解成了以下四个子分类问题的叠加：

1. 原本的反讽四分类问题。
2. “没有反讽”和“基于相反语义的反讽”二分类。
3. “没有反讽”和“情景反讽”二分类。
4. “没有反讽”和“其他反讽”二分类。

对于每个子分类问题，我们分别准备其对应的分类器组，依次命名为第一、第二、第三、第四组分类器，分别由 N_1、N_2、N_3、N_4 个分类器组成。那么对于一条待识别的微博，决策过程如下：

- 首先，由对应原四分类问题的第一组分类器进行多数投票，得出预测标签 $Label^1_{MV}$，直接以它作为第一步的预测标签 $Label_I$ 。以下把系统到这一步为止的判断结果称为中间结果 I。
- 第二步，由面向“没有反讽”和“基于相反语义的反讽”的第二组分类器进行多数投票，得出预测标签 $Label^2_{MV}$。若其中超过 thr_2 个分类器投票给 $Label^2_{MV}$ 且第一步的预测标签 $Label_I$ 为“没有反讽”或“基于相反语义的反讽”，则把预测标签修改为 $Label^2_{MV}$，否则保持不变，以此得出第二步的预测标签 $Label_{II}$。以下把系统到这一步为止的判断结果称为中间结果 II。
- 第三步，由面向“没有反讽”和“情景反讽”的第三组分类器进行多数投票，得出预测标签 $Label^3_{MV}$。若其中超过 thr_3 个分类器投票给 $Label^3_{MV}$ 且第二步的预测标签 $Label_{II}$ 为“没有反讽”或“情景反讽”，则把预测标签修改为 $Label^3_{MV}$，否则保持不变，以此得出第三步的预测标签 $Label_{III}$。以下把系统到这一步为止的判断结果称为中间结果 III。
- 最后一步，由面向“没有反讽”和“其他反讽”的第四组分类器进行多数投票，得出预测标签 $Label^4_{MV}$。若其中超过 thr_4 个分类器投票给 $Label^4_{MV}$ 且第三步的预测标签 $Label_{III}$ 为“没有反讽”或“其他反讽”，则把预测标签修改为 $Label^4_{MV}$，否则保持不变，以此得出第四步的预测标签 $Label_{IV}$，同时作为整个系统对该微博最终的反讽识别结果。

整个决策过程可以分成两大部分。第一部分的目的是初步完成对微博的四分类反讽识别，对应上述四步决策中的第一步。第二部分的目的是基于第一部分的初步识别结果逐步进行修正，每一步只关注一个子分类问题。对应上述四步决策

中的后三步，每步只关注“没有反讽”和其中一个反讽子类的二分类问题，由一组专门的分类器给出子分类问题的识别结果，当新的识别结果充分可信则修改前一步得到的预测标签。在这里新识别结果的可信度由投票数决定，当多数票是由超过 thr 个分类器投票所得则认为它充分可信。

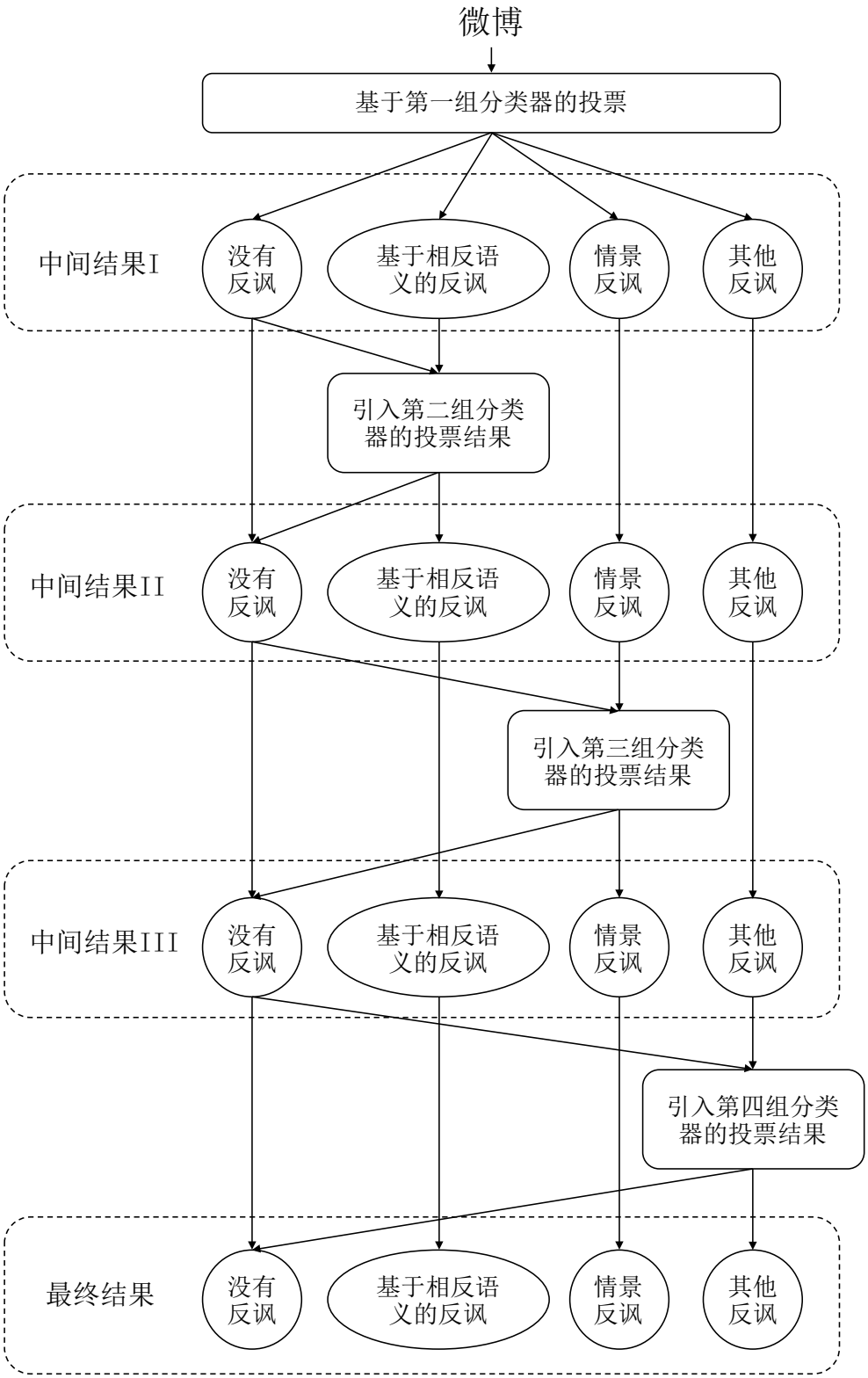


图 3.3　面向四分类反讽识别的系统框架

其中对于每个子分类问题，我们都采用了相同的模型框架，如图3.4所示，每个子分类器的输入都是微博文本经过预处理后得到的词序列 $\{w_i\}$，然后每个词替

换成对应的词嵌入向量，作为特征编码器的输入。特征编码器的目的是把微博文本对应的词向量序列转换成固定长度的特征向量，作为其反讽属性相关的表示向量。

特征编码器首先由一层或多层卷积神经网络或迭归神经网络组成，但二维卷积神经网络和迭归神经网络的输出均为和输入序列等长度的向量序列，我们需要进一步结合整个向量序列的信息并转换成固定长度的向量。对于卷积神经网络，主流做法之一是采到最大池化层，分别取各特征位上的最大值。而对于迭归神经网络，主流做法之一是取序列的最后一个向量，在理论上迭归神经网络在最后一时刻的输出结合了所有输入的信息，另一种是采用注意力机制。

接下来，我们将微博的表示向量作为概率预测器的输入，概率预测器的目的是得出各个反讽类别的概率分布。此处概率预测器采用单层的全联接层，以 $Softmax$ 作为激活函数得出概率分布，最后取概率最高者作为分类器对该条微博的预测结果。

考虑到对于不同反讽类别的语言特征不同，各个模型的建模能力也会有所不同。所以对于每个子分类问题，我们会分别比较各个模型的性能，以求在子分类问题上达到尽可能好的识别性能。

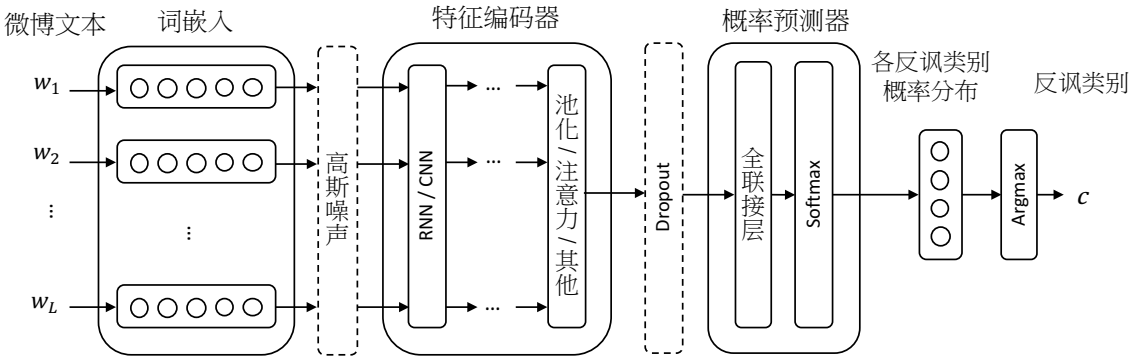


图 3.4　反讽识别子分类器模型框架

3.5　实验与分析

3.5.1　数据预处理

基于我们在章节3.3.4中对样本文本的观察，我们依次采取了以下数据预处理手法：

- 对于不用的用户标签“@someuser”，我们假设具体的用户名不影响微博内容的反讽类别，故统一替换成“<user>”。
- 对于井号标签“#something”，我们将其替换成一个字符串序列

“<hashtag>”、“something”、“</hashtag>”，此处“<hashtag>”表示井号标签的开始，“</hashtag>”表示井号标签的结束，原因在于 Twitter 平台上，井号标签可能由多个单词组成，如语料中出现的井号标签“#SoCute”，应该分解成“so”、“cute”两个单词，而语料中多单词组成的井号标签普遍采用首字母大写示意，故可以简单完成分词，同时在前后添加“<hashtag>”和“</hashtag>”示意中间的内容属于同一个井号标签即可。

- 对于全字母大写的文本段，在该段文本的前后添加“<allcap>”和“</allcap>”示意。如“YAYYYY”替换成序列“<allcap>”、“yayyyy”、“</allcap>”。
- 对于重复次数大于等次三次的标点符号，以“<repeated>”示意，如“!!!”替换成序列“!”、“<repeated>”，表示“!”被多次重复，即时假设重复的具体次数和原微博的反讽类别无关。
- 对于字母被故意重复的单词，以“<elongated>”示意，如“Noooooooo”替换成序列“No”、“<elongated>”，表示“No”中某一个或多个字母被多次重复，即假设被重复的字符和具体重复的次数和原微博的反讽类别无关。
- 将数字串替换成“<num>”，将电话号码替换成“<phone>”，将日期和时间分别替换成“<date>”和“<time>”，将数字百分比替换成“<percentage>”，超链接替换成“<url>”，即假设其中的具体数值和原微博的反讽类别无关。
- 对于由多个标点符号组成的表情符，我们将其替换成对应的情感标签，如将“:)”替换成“<happy>”，“:((”替换成“<sad>”。
- 在完成以上处理后，对英文的大小写统一转换成小写。

以上功能我们利用了第三方的英语文本处理工具 *ekphrasis*[①]完成，对于其中如何分词、如何识别电话号码和日期、以及颜文字到情感标签的详细映射关系列表，读者可以直接参考其代码实现和配置文件。

3.5.2　评价指标

按照国际比赛 SemEval-2018 任务三的设置，各个子任务均以 F1 值作为系统识别性能的主要评价指标。对于其中一个类别 c 的 F1 值，其定义如下：

$$F_c = \frac{2 \times P_c \times R_c}{P_c + R_c} \tag{3-1}$$

其中 P_c 为类别 c 的正确率，R_c 为类别 c 的召回率，其定义如下：

① https://github.com/cbaziotis/ekphrasis

$$P_c = \frac{TP_c}{TP_c + FP_c} \tag{3-2}$$

$$R_c = \frac{TP_c}{TP_c + FN_c} \tag{3-3}$$

其中 TP_c 表示被系统预测为类别 c，且真实标签为类别 c 的样本数量；FP_c 表示被系统预测为类别 c，但真实标签不是类别 c 的样本数量；FN_c 表示被系统预测为不是类别 c，但真实标签为类别 c 的样本数量。对于子任务一，系统性能以“带有反讽”一类样本的 F1 值为主要评估指标。对于子任务二，系统性能以各个类别的 F1 值的宏平均作为主要评价指标，即：

$$F_{macro} = \sum_{c \in C} F_c \tag{3-4}$$

此处 C 对应子任务二中四个类别组成的集合，即 {没有反讽，基于相反语义的反讽，情景反讽，其他反讽}。在以下实验中，除了 F1 值、正确率和召回率，我们还会观察模型的准确率，其定义如下：

$$Acc = \frac{\sum\limits_{c \in C} TP_c}{\sum\limits_{c \in C} (TP_c + FP_c)} \tag{3-5}$$

其中 TP_c 和 FP_c 如前述的定义，而 C 在子任务一中对应两个类别组成的集合，即 {没有反讽，带有反讽}。

3.5.3　模型训练

对于每个子分类问题，我们分别从完整的训练数据和测试数据中筛选出对应类别的样本作为实验数据。并采用相同的方法进行模型训练，其中涉及一些策略如下：

- 对于训练数据，我们分别从各个类别的样本中随机选出 90% 的样本作为训练集，剩余 10% 的样本作为验证集，以保留各个类别的样本分布。在每一轮模型参数调整后，计算各组模型参数在验证集上的 F1 值，经过有限轮迭代后，取在验证集上 F1 值最优的网络参数作为该分类器的参数，以此避免在训练数据上过拟合。

- 由于在我们的系统中，每个子分类问题由多个分类器经过多数投票给出识别结果，为了充分运用训练数据，在训练过程中每个分类器的验证集为独立随机筛选得出，一方面使得每个训练样本都有概率被用于某个分类器的模型训练，另一方面各个分类器的训练集不同，避免了对部分数据的过拟合。
- 对于词嵌入层，我们利用了章节2.4.2.1中 Baziotis 等人[47] 提供的词嵌入模型，直接用于初始化词嵌入层的参数。而对于词嵌入模型中未被覆盖的单词，若它在至少 2 个训练样本中出现，则为其随机生成词向量。在训练过程中，我们不对词嵌入层的参数进行调整。
- 在参数训练阶段，我们对词嵌入层输出的词向量序列添加高斯噪音。由于词嵌入算法在原理上使得意思相似的单词投影到词嵌入空间中距离相近的点上，高斯噪音的添加相当于把原本的单词替换成近义词，使得模型能更好地理解近义词构成的语言模式，另一方面缓解过拟合的问题。在验证阶段和测试阶段，高斯躁音的标准方差被调整为零，即不起作用。
- 在参数训练阶段，我们在特征编码器和概率预测器之间添加了 Dropout 层，以概率 $p_{Dropout}$ 将特征编码器得出特征向量上的各位数值置为零，并对没有被置零的各位数值乘以常量 $\frac{1}{1-p_{dropout}}$。在验证阶段和测试阶段，Dropout 层不起作用。
- 对于模型训练的损失函数，我们以权重 l_2 加入了概率预测器中全联接层的权重（不包括偏移量）的 L2 正则项。
- 在面向“没有反讽”和“其他反讽”的二分类问题中，由于两个类别的样本数据差异较大（1923：205），当人工神经网络根据样本的误差透过反向传播进行学习时，如果所有样本的权重相同，模型会倾向于把所有样本判断为样本量够多的“没有反讽”。为此我们根据样本量的分布决定各个类别的样本的权重。如公式3-6所示，其中 w_c 表示类别 c 对应每个样本的权重，N 表示总的训练样本数，C 表示该子分类问题的类别集合，此处即为 {没有反讽，其他反讽}，N_c 示类别 c 对应的训练样本数。

$$w_c = \frac{N}{|C| \times N_c} \tag{3-6}$$

3.5.4　实验结果与分析

对应章节 2.3，我们的实验分成两大部分。第一部分针对各个子分类问题进行研究，根据章节 3.4，我们的反讽识别系统涉及以下多个子分类问题：区分四个反

讽类别的四分类问题、区分“没有反讽”和“基于相反语义的反讽”的二分类问题、区分“没有反讽”和“情景反讽”的二分类问题、区分“没有反讽”和“其他反讽”的二分类问题。对于每个子分类问题，我们基于章节 3.4中提出的分类器模型框架进行实验，比较不同模型的建模能力，同时观察它们在不同子分类问题下性能区别。

第二部分对我们提出的基于多步决策的的识别系统进行研究。一方面评估我们系统的整体性能水平，另一方面透过观察系统的中间结果分析多步决策如何对系统的整体性能带来影响，从而解释此系统设计的合理性。

3.5.4.1　面向“没有反讽”和“带有反讽”二分类的模型性能分析

对面向“没有反讽”和“带有反讽”的二分类问题，表 3.5和图 3.5显示各个模型在测试集上的性能。注意表中的正确率、召回率、F1 值均是针对类别“带有反讽”的指标，对应比赛 SemEval2018 任务三子任务一关注的主要指标。

对于 F1 值，2 层 BiLSTM 的表现最好，其次是 LSTM 配合注意力机制，其 F1 值和前者非常接近（偏差 0.0009），第三为单层的 BiLSTM，其 F1 值和前两者则差距明显（0.0093 以上）。对于准确率，单层 BiLSTM 配合注意力机制和 2 层 BiLSTM 配合注意力机制达到最好的数值 0.6888 ，第三为 LSTM 配合注意力机制，其准确率和前两者偏差较小（0.026）。对于正确率，2 层 BiLSTM 配合注意力机制达到最好的数值 0.5870 ，其次是单层 BiLSTM 配合注意力机制，与前者偏差较小（0.0009），第三的 LSTM 配合注意力机制则和和前两者则差距明显（0.093）。对于召回率，单层的 LSTM 达到最好的 0.8617 ，第二的 2 层 BiLSTM 和前者的差距明显（0.0385）。

对于单层 BiLSTM 和 2 层 BiLSTM，添加注意力机制都使得准确率和“带有反讽”的正确率有所提升，但同时“带有反讽”的召回率明显下降，导致“带有反讽”的 F1 值也连带下降，可见 BiLSTM 和 2 层 BiLSTM 画添加注意力机制后整个模型的拟合能力是有上升的，但更倾向于预测样本为“没有反讽”，导致“带有反讽”的召回率下降，相对地更少比例的样本被误判为“带有反讽”，因而正确率稍为提高。另外 CNN 在添加注意力机制后，除召回率以外的三项指标都达到了各个模型中最差的性能，可见虽然在公式上注意力机制可以配合 CNN 使用，但实际性能并不理想。

表 3.5　面向“没有反讽”和“带有反讽”二分类的模型性能

	准确率	正确率	召回率	F1 值
CNN	0.6658 (6)	0.5548 (6)	0.7974 (5)	0.6544 (5)
CNN+ 注意力机制	0.6008 (12)	0.4980 (12)	0.8039 (3)	0.6150 (12)
GRU	0.6607 (7)	0.5542 (7)	0.7395 (9)	0.6336 (10)
GRU+ 注意力机制	0.6569 (10)	0.5482 (9)	0.7685 (7)	0.6399 (9)
BiGRU	0.6594 (8)	0.5534 (8)	0.7331 (10)	0.6307 (11)
BiGRU+ 注意力机制	0.6582 (9)	0.5473 (10)	0.8006 (4)	0.6501 (7)
LSTM	0.6390 (11)	0.5276 (11)	**0.8617 (1)**	0.6545 (4)
LSTM+ 注意力机制	0.6862 (3)	0.5768 (3)	0.7846 (6)	0.6640 (2)
BiLSTM	0.6798 (4)	0.5721 (4)	0.7653 (8)	0.6547 (3)
BiLSTM+ 注意力机制	**0.6888 (1)**	0.5861 (2)	0.7331 (10)	0.6514 (6)
2 层 BiLSTM	0.6709 (5)	0.5577 (5)	0.8232 (2)	**0.6649 (1)**
2 层 BiLSTM+ 注意力机制	**0.6888 (1)**	**0.5870 (1)**	0.7267 (12)	0.6494 (8)

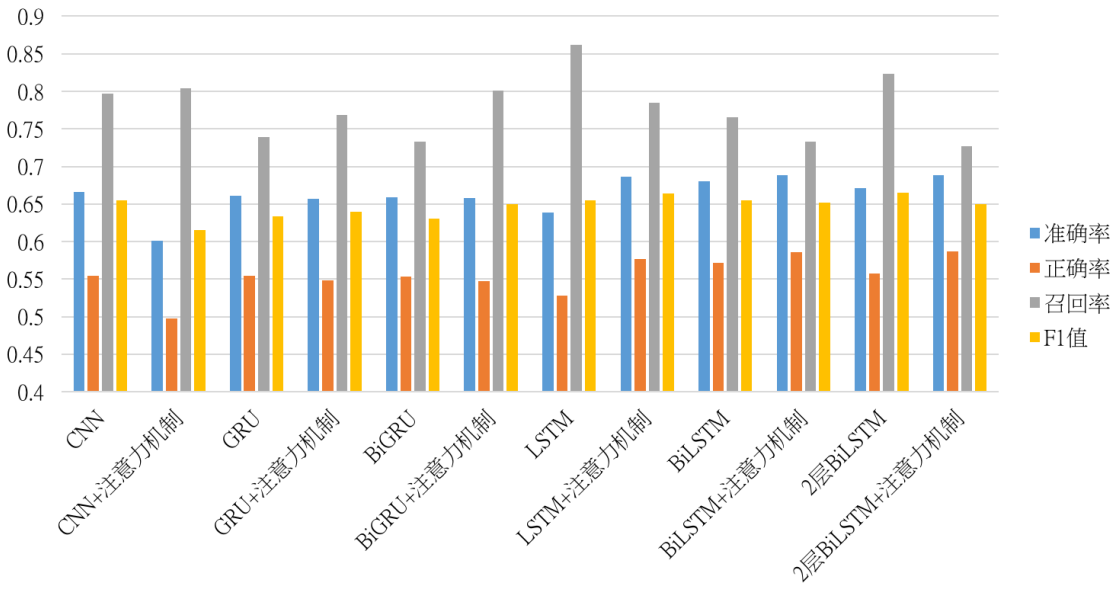


图 3.5　面向“没有反讽”和“带有反讽”二分类的模型性能

3.5.4.2　面向反讽四分类的模型性能分析

对于面向反讽四分类问题，表 3.6和图 3.6显示各个模型在测试集上的性能。注意表中的正确率、召回率、F1 值均是各类别对应指标数据的宏平均，对应比赛 SemEval2018 任务三子任务二关注的主要指标。

对于 F1 值，BiGRU 达到最好的数值（0.4768），其次是 2 层 BiLSTM 以及 2 层 BiLSTM 配合注意力机制，与前者的数值差距明显（约 0.0111）。对于准确率，同样由 BiGRU 达到最好的数值（0.6722），其次是 BiLSTM 和 2 层 BiLSTM，和前者

的数值差距明显（0.014 以上）。对于正确率，CNN 和 2 层 BiLSTM 的性能较好（在 0.5 以上），第三的 BiLSTM 则在 0.49 以下。对于召回率，2 层 BiLSTM 配合注意力机制达到最好的 0.4864，其次的 BiGRU 和 BiLSTM 则和前者差距明显（0.0085）。

另外，六组模型在添加注意力机制后正确率、召回率和 F1 值都有所下降（仅 2 层 BiLSTM 的召回率例外），而对于准确率较高的 BiGRU、BiLSTM 和 CNN 在添加注意力机制后准确率同样有所下降，可以认为添加注意力机制在此子分类问题中并不没有带来性能提升。

表 3.6　面向反讽四分类的模型性能

	准确率	正确率	召回率	F1 值
CNN	0.6531 (4)	**0.5090 (1)**	0.4667 (5)	0.4432 (7)
CNN+ 注意力机制	0.6186 (10)	0.4467 (7)	0.4214 (12)	0.4030 (12)
GRU	0.6148 (12)	0.4437 (8)	0.4693 (4)	0.4476 (5)
GRU+ 注意力机制	0.6531 (4)	0.4253 (10)	0.4605 (7)	0.4370 (8)
BiGRU	**0.6722 (1)**	0.4868 (4)	0.4779 (2)	**0.4768 (1)**
BiGRU+ 注意力机制	0.6301 (9)	0.4543 (6)	0.4638 (6)	0.4497 (4)
LSTM	0.6186 (10)	0.4313 (9)	0.4490 (9)	0.4307 (9)
LSTM+ 注意力机制	0.6416 (6)	0.4046 (12)	0.4394 (10)	0.4148 (11)
BiLSTM	0.6582 (2)	0.4875 (3)	0.4537 (8)	0.4447 (6)
BiLSTM+ 注意力机制	0.6352 (7)	0.4168 (11)	0.4386 (11)	0.4160 (10)
2 层 BiLSTM	0.6582 (2)	0.5068 (2)	0.4762 (3)	0.4644 (3)
2 层 BiLSTM+ 注意力机制	0.6314 (8)	0.4824 (5)	**0.4864 (1)**	0.4657 (2)

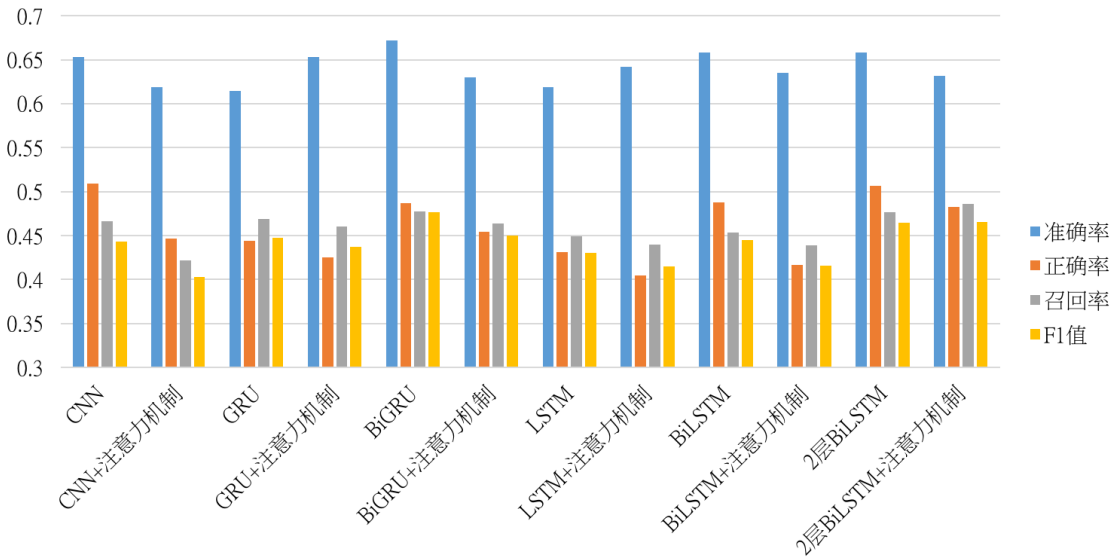


图 3.6　面向反讽四分类的模型性能

3.5.4.3　面向“没有反讽”和“基于相反语义的反讽”二分类的模型性能分析

对于面向“没有反讽”和“基于相反语义的反讽”的二分类问题，表 3.7和图 3.7显示各个模型在测试集上能达到的性能。注意表中的正确率、召回率、F1 值指“没有反讽”和“基于相反语义的反讽”两个类别对应指标的宏平均。

对于 F1 值，LSTM 配合注意力机制达到最大数值的 0.7752 ，其后依次为 BiGRU 和 GRU，和前者差距约较小（约 0.0047)。对于准确率，同样是 LSTM 配合注意力机制达到最好的效果（0.8275），其后依次为 GRU 和 BiGRU，和前者差距大（0.0157)。对于正确率，还是 LSTM 配合注意力机制的数值最高（0.7717），其后依次为 GRU 和 BiGRU，和前者差距较大（0.0151）。对于召回率，数值最高的是 BiGRU（0.7993），其次是 GRU 和 LSTM，和前者差距均较小（约 0.0049）。

总的来说，LSTM 配合注意力机制在四项指标中的三项都达到了最好的性能。GRU 和 BiGRU 的性能非常接近，各项指标在所有模型中排第二和第三。从数学模型上看，BiGRU 比单层 GRU 多一个把文本反向输入的 GRU 通道，提高了召回率的同时稍微降低了准确率和正确率，显示反向输入的 GRU 通道能额外捕足到“基于相反语义的反讽”的相关特征。但相对地 BiLSTM 在四项指标上都明显低于 LSTM，我们认为其中的原因可能有两点，一是 LSTM 本身对“基于相反语义的反讽”的建模能力较差，二是 BiLSTM 的模型参数过多导致对训练集的过拟合。

表 3.7　面向“没有反讽”和“基于相反语义的反讽”的二分类各模型性能

	准确率	正确率	召回率	F1 值
CNN	0.7991 (5)	0.7438 (5)	0.7791 (7)	0.7562 (5)
CNN+ 注意力机制	0.6829 (12)	0.6494 (12)	0.6909 (12)	0.6470 (12)
GRU	0.8100 (2)	0.7566 (2)	0.7944 (2)	0.7699 (3)
GRU+ 注意力机制	0.7677 (9)	0.7196 (9)	0.7679 (10)	0.7303 (9)
BiGRU	0.8085 (3)	0.7565 (3)	**0.7993 (1)**	0.7705 (2)
BiGRU+ 注意力机制	0.7755 (8)	0.7301 (8)	0.7831 (5)	0.7412 (8)
LSTM	0.8053 (4)	0.7525 (4)	0.7932 (3)	0.7661 (4)
LSTM+ 注意力机制	**0.8257 (1)**	**0.7717 (1)**	0.7791 (7)	**0.7752 (1)**
BiLSTM	0.7551 (10)	0.7169 (10)	0.7734 (9)	0.7236 (10)
BiLSTM+ 注意力机制	0.7174 (11)	0.6909 (11)	0.7460 (11)	0.6889 (11)
2 层 BiLSTM	0.7849 (7)	0.7339 (7)	0.7795 (6)	0.7465 (7)
2 层 BiLSTM+ 注意力机制	0.7928 (6)	0.7421 (6)	0.7888 (4)	0.7554 (6)

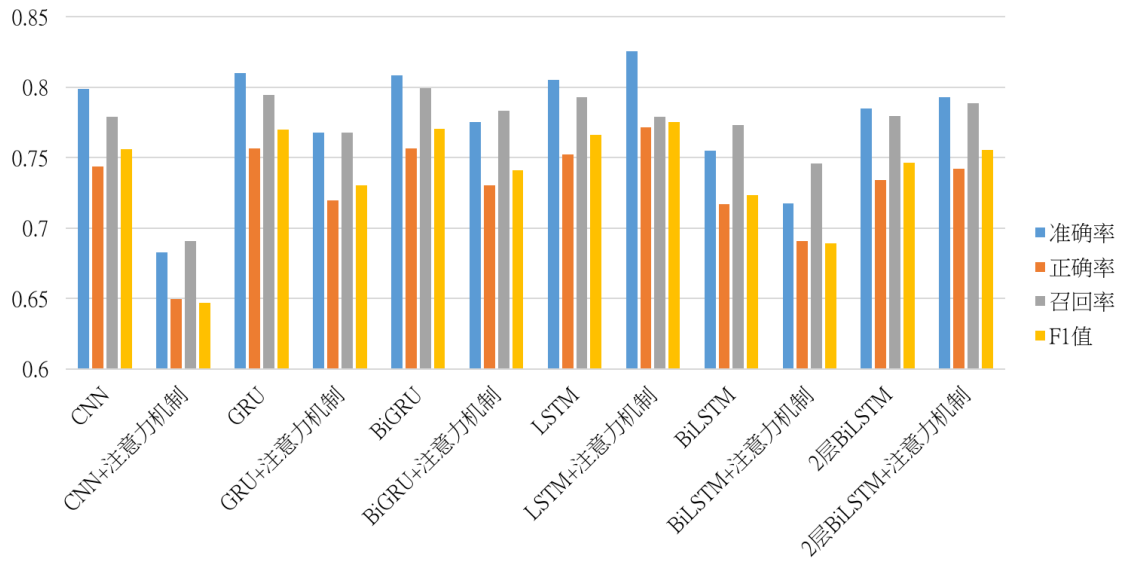


图 3.7　面向“没有反讽”和“基于相反语义的反讽”的二分类各模型性能

3.5.4.4　面向“没有反讽”和“情景反讽”二分类的模型性能分析

对于面向“没有反讽”和“情景反讽”的二分类问题，表 3.8和图 3.8显示各个模型在测试集上的性能。注意表中的正确率、召回率、F1 值指“没有反讽”和“情景反讽”两个类别对应指标的宏平均。

对于 F1 值和召回率，2 层 BiLSTM 配合注意力机制在这两个指标上都达到了最好的效果（数值分别为 0.6924 和 0.6806），其次是 2 层 BiLSTM，第三是单层的 BiLSTM 配合注意力机制，但后两者在准确率和正确率这两项指标上都排在靠后的位置。而对于准确率和正确率，均由 CNN 达到最好的效果（数值分别为 0.8656 和 0.7634），其次是 2 层 BiLSTM 配合注意力机制，第三是单层的 LSTM。

整体上，2 层 BiLSTM 配合注意力机制在四项指标上的都达到了靠前的效果，显示 2 层 BiLSTM 配合注意力机制对区分“没有反讽”和“情景反讽”有明显较好的建模能力。另外 CNN 虽然在准确率和正确率上都达到最好的性能，但由于召回率数值太低导致了 F1 值明显低于第一的 F1 值（差距约 0.045），显示 CNN 只对部分“情景反讽”的模式有较好的建模能力，所以识别为“情景反讽”的样本基本正确，但召回的样本不多。

表 3.8　面向“没有反讽”和“情景反讽”的二分类模型性能

	准确率	正确率	召回率	F1 值
CNN	**0.8656 (1)**	**0.7634 (1)**	0.6167 (7)	0.6473 (7)
CNN+ 注意力机制	0.8477 (5)	0.6865 (5)	0.5917 (11)	0.6117 (11)
GRU	0.8495 (4)	0.6942 (4)	0.6024 (10)	0.6242 (10)
GRU+ 注意力机制	0.8333 (11)	0.6708 (8)	0.6556 (4)	0.6625 (4)
BiGRU	0.8423 (6)	0.6645 (11)	0.5789 (12)	0.5951 (12)
BiGRU+ 注意力机制	0.8423 (6)	0.6825 (6)	0.6416 (5)	0.6572 (6)
LSTM	0.8513 (2)	0.7027 (3)	0.6372 (6)	0.6590 (5)
LSTM+ 注意力机制	0.8387 (8)	0.6676 (10)	0.6153 (8)	0.6325 (8)
BiLSTM	0.8351 (10)	0.6575 (12)	0.6084 (9)	0.6243 (9)
BiLSTM+ 注意力机制	0.8369 (9)	0.6797 (7)	0.6674 (3)	0.6731 (3)
2 层 BiLSTM	0.8262 (12)	0.6691 (9)	0.6803 (2)	0.6743 (2)
2 层 BiLSTM+ 注意力机制	0.8513 (2)	0.7076 (2)	**0.6806 (1)**	**0.6924 (1)**

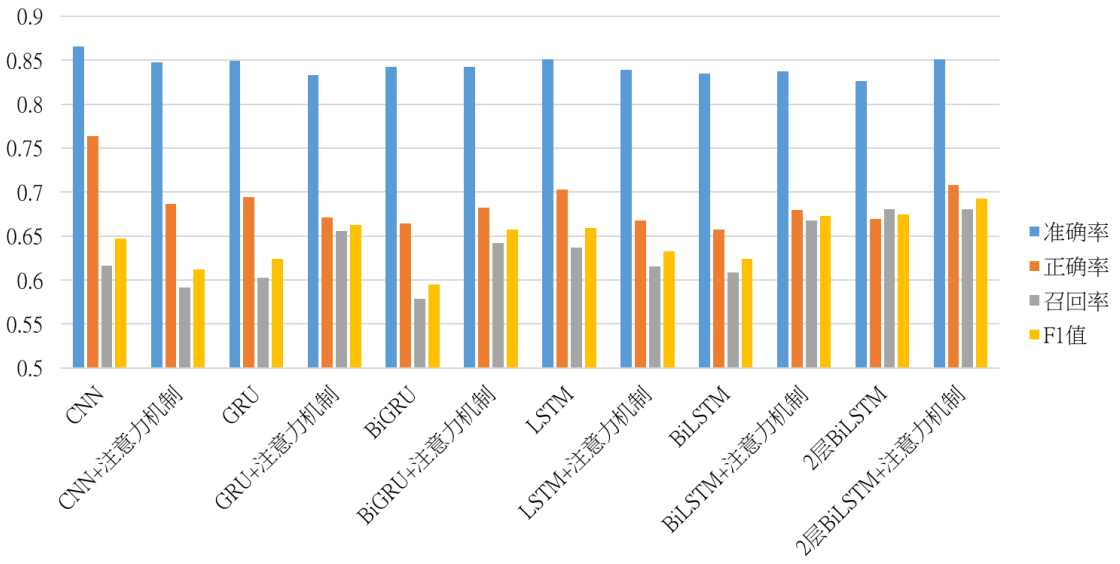


图 3.8　面向“没有反讽”和“情景反讽”的二分类模型性能

3.5.4.5　面向“没有反讽”和“其他反讽”二分类的模型性能分析

对于面向“没有反讽”和“其他反讽”的二分类问题，表 3.9和图 3.9显示各个模型在测试集上的性能。注意表中的正确率、召回率、F1 值指“没有反讽”和“其他反讽”两个类别对应指标的宏平均。

对于 F1 值，性能最好的是 CNN（0.6016），其次是 GRU，与前者差距明显（约 0.0163），第三为 BiLSTM，明显低于 CNN 的数值（差距约为 0.0489）。对于准确率，同样是 CNN 达到最高的数值（0.8860），其次是 GRU 配合注意力机制以及

LSTM 配合注意力机制，在数值上比较接近（差距仅 0.0019）。对于正确率，依然是 CNN 的性能最优（0.7123），第二是 GRU 配合注意力机制，但和前者差距较明显（约 0.0195），第三的 GRU 则远差于前两者（差距达 0.1 以上)。对于召回率，则是 GRU 的效果最好（0.5836），其次是 CNN，与前者差距较小（约 0.0055），第三的 BiLSTM 则和前两者差距较大（达 0.0313）。

整体上，CNN 在三项指标上都达到了最好的效果，而在召回率也逼近最好的 GRU，显示 CNN 对“其他反讽”有明显较好的建模能力。对比面向“没有反讽”和“情景反讽”的二分类实验，CNN 在准确率和正确率上同样达到最好的效果，但召回率和 F1 值都排在靠后的位置，可见相对于“情景反讽”，CNN 对“其他反讽”的建模能力更好。

表 3.9　面向“没有反讽”和“其他反讽”的二分类模型性能

	准确率	正确率	召回率	F1 值
CNN	**0.8860 (1)**	**0.7123 (1)**	0.5781 (2)	**0.6016 (1)**
CNN+ 注意力机制	0.7981 (11)	0.4930 (8)	0.4934 (11)	0.4932 (7)
GRU	0.8336 (10)	0.5873 (3)	**0.5836 (1)**	0.5853 (2)
GRU+ 注意力机制	0.8841 (2)	0.6928 (2)	0.5070 (7)	0.4848 (8)
BiGRU	0.8411 (8)	0.5481 (7)	0.5317 (6)	0.5353 (6)
BiGRU+ 注意力机制	0.8822 (4)	0.4419 (11)	0.4989 (9)	0.4687 (10)
LSTM	0.8486 (6)	0.5603 (5)	0.5360 (4)	0.5409 (4)
LSTM+ 注意力机制	0.8841 (2)	0.4421 (10)	0.5000 (8)	0.4692 (9)
BiLSTM	0.8430 (7)	0.5663 (4)	0.5468 (3)	0.5527 (3)
BiLSTM+ 注意力机制	0.8822 (4)	0.4419 (11)	0.4989 (9)	0.4687 (10)
2 层 BiLSTM	0.8355 (9)	0.5483 (6)	0.5356 (5)	0.5393 (5)
2 层 BiLSTM+ 注意力机制	0.7944 (12)	0.4571 (9)	0.4633 (12)	0.4600 (12)

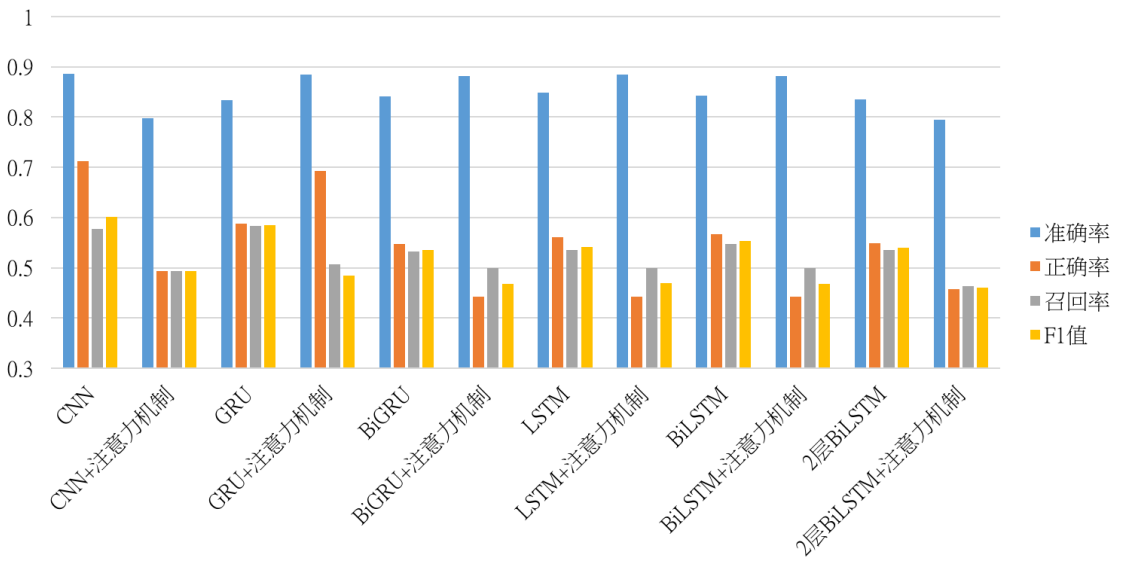


图 3.9　面向“没有反讽”和“其他反讽”的二分类模型性能

3.5.4.6　面向“没有反讽”和“带有反讽”二分类的系统性能分析

经过前面各章节的分析，我们已经对各个子分类问题中不同模型的性能有大致了解，接下来我们将研究由多组子分类器集成得出的反讽识别系统的性能，并和比赛 SemEval-2018 任务三的参赛系统比较来评估我们系统的性能水平。

首先对于 SemEval-2018 任务三的子任务一，即面向“没有反讽”和“带有反讽”的二分类问题，基于章节 3.5.4.1的实验结果，我们最终选择以 F1 值最高的 2 层 BiLSTM 作为子分类器的模型，按照章节 3.5.3训练出 9 个分类器组成分类器组，按照章节 3.4，我由该分类器组进行多数投票来判断一条微博是否带有反讽。

表 3.10显示我们的系统和 SemEval-2018 任务三子任务一中其他参赛系统在测试集上的性能。注意表中的正确率、召回率、F1 值均是针对类别“带有反讽”的指标。在 SemEval-2018 任务三子任务一的最终测试阶段，参赛系统共 43 个，表中仅显示排名前 10（按照 F1 值排名）或在某项指标排名靠前的参赛系统，其中队伍名称为 THU_HCSI 对应我们当时提交的系统，排名第 8。

首先观察各项指标上我们系统的性能水平。对于 F1 值，我们系统的数值为 0.7010，对应当时排名第二，略低于第一名的 0.7050，而显着高于当时第二名的 0.6719(差距约 0.03)。对于准确率，我们系统的数值为 0.7256，对应当时排名第三，和第一名的差距约为 0.009。对于正确率，我们系统的数值为 0.6176，对应当时排名第五。对于召回率，我们系统的数值为 0.8103，对应当时排名第四。总的来说我们的系统在上述四项指标上都达到了当时排名靠前的性能，对比我们当时提交的系统有了明显的提升。

另外对比单个子分类器（即单个 2 层 BiLSTM 模型的分类器）和整个系统的识别性能，系统在准确率、正确率和 F1 值上都有明显的提升，而召回率有轻微的下降，显示采用多数投票方法有效基于多个子分类器达到更好的识别性能。

表 3.10　SemEval-2018 任务三子任务一参赛系统性能

排名	队伍名称	准确率	正确率	召回率	F1 值
1	THU_NGN	**0.7347 (1)**	0.6304 (4)	0.8006 (4)	**0.7054**
2	NTUA-SLP	0.7321 (2)	0.6535 (2)	0.6913 (13)	0.6719
3	WLV	0.6429 (15)	0.5317 (20)	0.8360 (2)	0.6500
4	(无)	0.6607 (10)	0.5506 (13)	0.7878 (7)	0.6481
5	NIHRIO, NCL	0.7015 (3)	0.6091 (5)	0.6913 (13)	0.6476
6	DLUTNLP-1	0.6276 (19)	0.5199 (23)	0.7974 (5)	0.6294
7	ELiRF-UPV	0.6110 (23)	0.5059 (27)	0.8328 (3)	0.6294
8	**THU_HCSI**	0.6594 (11)	0.5550 (11)	0.7138 (10)	0.6245
9	CJ	0.6671 (8)	0.5654 (9)	0.6945 (12)	0.6234
10	#NonDicevoSulSerio	0.6786 (7)	0.5831 (8)	0.6656 (15)	0.6216
15	(无)	0.5651 (31)	0.4731 (33)	**0.8489 (1)**	0.6076
43	INGEOTEC-IIMAS	0.6276 (19)	**0.8800 (1)**	0.0707 (37)	0.1310
	单个分类器	0.6709 (8)	0.5577 (10)	0.8232 (4)	0.6649 (3)
	我们的系统	0.7258 (3)	0.6176 (5)	0.8103 (4)	0.7010 (2)

3.5.4.7　面向反讽四分类的系统性能分析

接下来我们分析面向反讽四分类的系统性能，对应 SemEval-2018 任务三的子任务二。按照章节 3.4，我们的反讽四分类系统由四子分类问题对应的分类器组所组成。对于和原问题相同的反讽四分类，根据章节 3.5.4.2的实验结果，我们采用 BiGRU 作为第一组分类器的模型。对于面向“没有反讽”和“基于相反语义的反讽”的二分类问题，按照章节 3.5.4.3的实验结果，我们采用 LSTM 配合注意力机制作为第二组分类器的模型。对于“没有反讽”和“情景反讽”的二分类问题，按照章节 3.5.4.4的实验结果，我们采用 2 层 BiLSTM+ 注意力机制作为第三组分类器的模型。对于“没有反讽”和“其他反讽”的二分类问题，按照章节 3.5.4.5的实验结果，我们采用 CNN 作为第四组分类器的模型。按照章节 3.5.3，我们基于选定的模型分别为每个分类器组训练 5 个分类器。

最后按照章节 3.4中给出的识别流程，透过逐步结合各组分类器的投票结果得出微博的反讽类别。对于系统中第二、三、四组分类器需要配置的参数 thr_2、thr_3 和 thr_4，我们均设置为 3，即对于每个子分类问题，直接按照多数投票的结果调整

预测标签。

表 3.11显示我们的系统和 SemEval-2018 任务三子任务二中其他参赛系统在测试集上的性能。注意表中的正确率、召回率、F1 值都是针对各类别对应指标的宏平均。在 SemEval-2018 任务三子任务二的最终测试阶段，参赛系统共 31 个，表中仅显示排名前 10（按照 F1 值排名）的参赛系统，而我们当时并未参与子任务二。

首先观察各项指标上我们系统的性能水平。对于 F1 值，我们系统的数值为 0.5205，对应当时排名第一，显着高于当时第一名的 0.5074。对于准确率，我们系统的数值为 0.6939，对应当时准确率排名第二，明显低于最高数值的 0.7321。对于正确率，我们系统的数值为 0.6003，对应当时正确率排名第一，显着高于当时最高 0.5768 的正确率。对于召回率，我们系统的数值为 0.5241，对应当时召回率排名第二，明显低于最高数值的 0.5414。

总的来说，我们的系统在各项指标上都达到了靠前的水平，而且在主要评价指标 F1 值上显着超过了当时的第一名。对比排名第一的系统，我们系统的准确率较低，而正确率和召回率都比它高，可以推断我们的系统成功召回了较多“带有反讽”的细分类别的样本，但同时把较多真实标签为“没有反讽”的样本误判为“带有反讽”的细分类别。对于准确率较低但宏平均 F1 值较高这一点，在数据不均匀的多分类问题中是个常见的情况。因为当系统对样本量较少的类别有较好的识别能力时，对样本量较多的类别的识别能力会相对较差，导致整体准确率下降。在数据不均匀的情况下各项性能指标一般不可兼得，因此在具体应用场景下需要慎重选择评价指标，并在系统开发时作针对性的设计。

表 3.11　SemEval-2018 任务三子任务二参赛系统性能

排名	队伍名称	准确率	正确率	召回率	F1 值
1	(无)	**0.7321 (1)**	**0.5768 (1)**	0.5044 (4)	0.5074
2	NTUA-SLP	0.6518 (4)	0.4959 (4)	0.5124 (2)	0.4959
3	**THU_NGN**	0.6046 (9)	0.4860 (6)	**0.5414 (1)**	0.4947
4	(无)	0.6033 (10)	0.4660 (7)	0.5058 (3)	0.4743
5	NIHRIO, NCL	0.6594 (3)	0.5446 (2)	0.4475 (5)	0.4437
6	Random Decision Syntax Trees	0.6327 (6)	0.4868 (5)	0.4388 (8)	0.4352
7	ELiRF-UPV	0.6327 (6)	0.4123 (12)	0.4404 (7)	0.4211
8	WLV	0.6709 (2)	0.4311 (10)	0.4149 (9)	0.4153
9	#NonDicevoSulSerio	0.5446 (18)	0.4087 (15)	0.4410 (6)	0.4131
10	INGEOTEC-IIMAS	0.6441 (5)	0.5017 (3)	0.3850 (15)	0.4055
	我们的系统	0.6939 (2)	**0.6003 (1)**	0.5241 (2)	**0.5205 (1)**

另外，由于我们的系统在每步决策后都可以作为一组预测结果，因此我们观察了每个中间结果在测试集上的性能。表 3.12显示我们系统的中间结果在测试集上的各项指标，注意表中的正确率、召回率、F1 值均是针对各类别对应指标的宏平均。

表中第一行的“中间结果 I”对应直接由第一组四分类器进行多数投票的识别结果，第二行“中间结果 II”是基于第二组分类器的投票对原本被识别为“没有反讽”和“基于相反语义的反讽”的样本重新进行二分类的识别结果，可见各项指标都有所提升。再对比中间结果 I 和中间结果 II 的混淆矩阵（分别对应表 3.13和表 3.14）可以发现正确召回了 9 个“没有反讽”的样本，同时误判了 3 个原本识别正确的“基于相反语义的反讽”的样本，故整体准确率还是有所上升。

表中第三行的“中间结果 III”是基于第三组分类器的投票对原本被识别为“没有反讽”和“情景反讽”的样本重新进行二分类的识别结果，对比“中间结果 II”的准确率不变，正确率稍微下降，同时召回率有了明显提高，导致 F1 值也明显提高。再对比中间结果 II 和中间结果 III 的混淆矩阵（分别对应表 3.14和表 3.15）可以发现误判了 14 个原本识别正确的“没有反讽”的样本，同时正确召回了 14 个“情景反讽”的样本，所以准确率不变，但由于“情景反讽”的样本量较少而“没有反讽”的样本量较多，“情景反讽”的召回率显着上升拉高了宏平均的召回率。这也显示了对于宏平均的指标，关注样本量少的类别能在数值上带来明显提升。

表中最底一行的“最终结果”是基于第四组分类器的投票对原本被识别为“没有反讽”和“其他反讽”的样本重新进行二分类的识别结果，可见各项指标都有所提高。再对比中间结果 III 和最终结果的混淆矩阵（分别对应表 3.15和表 3.16）可以发现这一步把 5 个被误判为“其他反讽”的样本改成了“没有反讽”，虽然其中只有 2 个样本的真实标签为“没有反讽”，但由于被识别为“其他反讽”的样本少，明显提高了“其他反讽”的正确率。

总结以上分析结合，我们基于多步决策的反讽识别流程透过逐步回答子分类问题有效得到更好的识别性能。我们认为其中的核心原因有以两点：

- 针对子分类问题进行建模。除了第一步决策对应原本的四分类问题，另外三个子分问题只关注其中的两个类别，只针对部分数据进行建模使得对应分类器对特定类别的识别能力更好。另外从章节 3.5.4.3、3.5.4.4、3.5.4.5可以看出，虽然“基于相反语义的反讽”、“情景反讽”和“其他反讽”都是“带有反讽”的细分类别，但在各个子分类问题上，达到较好识别性能的模型不同，显示这些类别背后的语言特征也不同。因此我们对于不同子分类问题分别选择合适的模型，再透过多步决策结合不同模型的结果，使得系统的整体识别能力比

只使用单种模型的效果更好。

- 针对样本不均匀的情况选择子分类问题。原本的四分类问题可以被拆解成多种子分类问题的叠加，但我们的系统里只考虑了“没有反讽”和各个反讽子类两两组合的二分类问题，原因在于“没有反讽”一类的样本量显着多于其他类别。从中间结果 I 对应的混淆矩阵（对应表 3.13）也可以看出这些类别最容易被误判为“没有反讽”，因此针对“没有反讽”和其他类别之间的识别结果进行调整能显着带来性能提升。

表 3.12　四分类反讽识别系统最终识别结果和中间结果的性能

	准确率	正确率	召回率	F1 值
中间结果 I	0.6837	0.5606	0.4891	0.4913
中间结果 II	0.6913	0.5655	0.4893	0.4949
中间结果 III	0.6913	0.5587	0.5231	0.5179
最终结果	**0.6939**	**0.6003**	**0.5241**	**0.5205**

3.5.5　错误分析

本节中，我们将对系统错误识别的样本进行观察，分析其中的原因，从而对问题有更探入的了解，以及尝试给出下一步的改进方向。以下使用的例子均参照表 3.17。

- **词组语义倾向识别**。参考例子 1，其中“blame”表示指责，在字面意思上有明显的负面情感倾向，而“XXX of the Year”意思为年度最佳的某某，整个词组带有正面情感倾向，故和前者对比得出原微博带有“基于相反语义的反讽”。关键在于词组“XXX of the Year”在逐字解读时没有明显的情感倾向，这也是文本情感识别中常见的问题之一。那么要使得基于人工神经网络的系统有能力识别词组的语义倾向，其必要条件之一是在训练集中遇到过类似的词组，对此可以透过收集更多训练样本来保证对常用词组的复盖率，但对于不常用或

表 3.13　反讽四分类测试集上中间结果 I 对应的混淆矩阵

		预测标签			
		没有反讽	反义反讽	情景反讽	其他反讽
真实标签	没有反讽	380	73	13	7
	反义反讽	36	125	3	0
	情景反讽	44	13	25	3
	其他反讽	41	10	5	6

表 3.14　反讽四分类测试集上中间结果 II 对应的混淆矩阵

		预测标签			
		没有反讽	反义反讽	情景反讽	其他反讽
真实标签	没有反讽	389	64	13	7
	反义反讽	39	122	3	0
	情景反讽	47	10	25	3
	其他反讽	39	12	5	6

表 3.15　反讽四分类测试集上中间结果 III 对应的混淆矩阵

		预测标签			
		没有反讽	反义反讽	情景反讽	其他反讽
真实标签	没有反讽	375	64	27	7
	反义反讽	35	122	7	0
	情景反讽	33	10	39	3
	其他反讽	38	12	6	6

新出现的词组则只能额外引入语言知识。

- **场景描述识别**。参考例子 2，用户表示他在担任校医当天发烧了，其中“担任校医”为场景，“发烧”的发生和对“担任校医”的预期相反，因此属于“情景反讽”，但“fever”本身间接带有负面情感，该样本被我们的系统误判成了“基于相反语义的反讽”。同样地对于例子 3，用户表示工作怎么能在他没有上班的期间完成，其中“没有上班”为场景，“工作完成”的发生和对“没有上班”的预期相反，因此属于“情景反讽”，但“off week”间接带有正面情感，因而被系统为识别为“基于相反语义的反讽”。再观察我们系统在测试集上的混淆矩阵（表 3.16），可以发现有相当一部分“情景反讽”的样本被误判为“基于相反语义的反讽”，我们由此推断其中的原因之一在于部分“情景反讽”的样本中确实包含带有情感极性的部分，因而导致了混淆，相对地专注于识别“场景”的出现会是召回这部分样本的关键。

表 3.16　反讽四分类测试集上最终识别结果对应的混淆矩阵

		预测标签			
		没有反讽	反义反讽	情景反讽	其他反讽
真实标签	没有反讽	377	64	27	5
	反义反讽	35	122	7	0
	情景反讽	36	10	39	0
	其他反讽	38	12	6	6

- **一词多义**。参考例子 4，“#Starbucks”是一家国际连锁咖啡店的名字，而在咖啡店的场景下，“tall”指咖啡杯一种大小，“blonde”为该店某个咖啡系统列的名称，而在日常生活场景中，“tall blonde”还有高挑金发女子的意思，即出现一词多义，而后者的意思在咖啡店场景下发生就产生了和预期相反的效果，因此属于“情景反讽”。换言之，要识别出例子中的“情景反讽”就要求系统认识到“tall blonde”具有咖啡以外的意思。由于在我们的模型中，每个单词都以一个固定的词向量表示，因此无法表示一词多义，更无法对多种含意的词组进行建模，在模型中考虑一词多义是解决方法之一。另外，让系统能够将“Starbucks”和咖啡关联起来则要求在训练集上有包含对应信息的样本。
- **信息不足**。参考例子 5，“clashupdate”并非标准英语单词或网络上的常用单词，后面紧接着一个超链接，仅凭字面意思无法理解用户要表达的意思。而访问该链接对应网页会发现插图为游戏“Clash of Clans”的更新提示，所以文本中的“clashupdate”应对应“clash update”，依然没有表示反讽的意思，但我们发现原微博的完整文本内容为“clashupdate #not”，其中“#not”在微博反讽识别的研究中是最常被用于对微博进行自动标注的井号标签之一，被组织者在准备数据集时故意过滤了。在比赛中，尝试找到微博原文有悖于组织者的初衷（不依赖井号标签进行分类），但在具体应用场景中，若要正确判断这些样本就可以考虑引入链接中的评论，这也对应下一章中探讨引入上下文的情感识别。

表 3.17　反讽识别子任务二中被系统误判的样本例子

编号	类别	微博
1	反义反讽	I blame my mom. Mother of the Year
2	情景反讽	So I’m the school nurse today. And I have a fever.
3	情景反讽	Also it is funny how things at work get done when I’m on my off week
4	情景反讽	Just walked in to #Starbucks and asked for a “tall blonde”Hahahaha
5	其他反讽	clashupdate http://t.co/1bFOsIwnI5

3.6　本章小结

在本章中我们提出了一种基于多步决策的多分类系统设计方案，并应用于面向微博的反讽识别。我们基于国际比赛 SemEval-2018 任务三的两个子任务进行了多组实验。对于子任务一，即“没有反讽”和“带有反讽”二分类，我们的模型达到了当时靠前的效果，和我们当时提交的系统性能相比有了显着的提升。对于子任务二，即“没有反讽”、“基于相反语义的反讽”、“情景反讽”和“其他反讽”四分类，基于

我们提出的系统设计方案，我们给出了一个具体的基于多步决策的反讽识别系统，实验结果显示我们系统的性能在当时的参赛系统中排名第一。另外，基于对系统中间结果的分析，我们验证了多步决策中每一步的有效性，并解释它如何对系统的整体性能带来影响。最后我们对被系统错误识别的样本进行分析，指出系统的不足之处并给出对应的改进方向。

第 4 章　基于多通道模型引入上下文的情感识别

4.1　本章引论

在面向短文本的情感识别场景，有时候仅凭一段文本的内容可能无法完全理解发言者想表达的内容。考虑在一个甲乙两人对话的场景中，甲说“彼此彼此！”，我们可以认为甲表达了和乙相同的想法，但我们无法确认甲表达的情感。假如乙原本说“庆喜庆喜!”，那么可以认为甲也在表示祝贺，表达的是一种正面的情感。但假如乙原本说的是“你也不过如此！”，那么可以为甲在表示不满，表达的是一种负面的情感。因此在情感识别中，一些研究会考虑引入上下文信息作为辅助以提高识别性能。譬如 Zahiri 和 Choi[39] 在研究电影剧剧本中每句台词的情感识别时，就引入了每句台词前的有限句台词作为上下文信息。Kunwoo 等人[75] 在面向两人对话的情感识别研究中，同样引入了当前发言之前的有限段对话作为上下文信息。然而这些研究工作对上下文的建模方式都有所不同，如在 Zahiri 和 Choi[39] 的研究中，一段剧本可能涉及多个角色，但在他们的模型中并没有考虑上下文中每句台词对应的角色，而在 Kunwoo 等人[75] 的研究中，由于对话过程只涉及两个人，在他们的模型中就区分了上下文中不同发言者说的话。对于不同场景下，如何在算法建模中引入上下文信息始终没有一种固定的方法，这也是因为不同类型的上下文包含的信息不同，对应具体的问题，我们依然需要进行针对性的模型设计。

国际比赛 SemEval-2019 的任务三[2] 则是旨在促进引入上下文的文本情感识别研究，比赛要求参赛者开发一个情感识别系统，对三轮对话中最后一轮发言表达的情感进行分类，给定的四个情感类别包括：开心、悲伤、愤怒、其他。为此我们提出了一个多通道模型，以引入作为上下文的前两轮对话。本章节中我们将基于 SemEval-2019 的任务三进行实验，采用比赛组织者提供的训练数据和测试数据，并透过和其他参赛系统进行比较来评估我们系统的识别性能。

本章的内容安排如下。在章节 4.2中，我们会首先给出当前问题的形式化表示。在章节 4.3中我们再对具体实验数据进行观察，分析给定数据集中各个情感类别的分布情况以及其文本特征等。在章节 4.4将给出我们针对当前问题提出的识别系统框架，以及组成该系统的多通道模型结构。最后在章节 4.5，我们会给出实验的细节，以及对实验结果进行分析。

4.2　形式化表示

在本章中，我们将研究面向三轮对话的情感识别，以下我们对应章节 2.2给出此问题的形式化表示。给定一个情感类别集合 C，对于一个三轮对话的集合 S，其中任意一个元素 s 可以表示为一个三元组 $< t^1, t^2, t^3 >$，三元组中的元素依次对应每轮发言的文本内容。而在上下文为 $b =< t^1, t^2 >$ 的情况，最后一轮发言 t^3 所表达的情感属于唯一一种情感类别 $c \in C$。又给定一个词集合 W，对任意一轮发言的文本 $t^j, j = 1, 2, 3$，经过文本预处理后可以表示为一个长度为 L^j 的词序列 $w^j =< w_1^j, w_2^j, ..., w_{L^j}^j >, w_i^j \in W, i \in [1, L^j]$。那么我们的目标是找出一个映射关系 F_C，使得 $c = F_C(w^3, < w^1, w^2 >)$。

4.3　实验数据

我们的实验采用 SemEval-2019 的任务三提供的数据集，其中每个样本对应一个三轮对话以及第三轮发言的情感标签，第一轮为用户甲的发言，第二轮为用户乙对第一轮的回复，第三轮为用户甲对第二轮中用户乙的回复。情感标签对应四个情感类别中的其中一种：开心、悲伤、愤怒、其他。表 4.1显示数据集各类别样本数量分布，表 4.2为语料中各个类别对应的样本例子。

表 4.1　情感识别各类别样本数量分布

数据集	其他	开心	悲伤	愤怒
训练集	14948	4243	5463	5506
验证集	2338	142	125	150
测试集	4677	284	250	298

在训练集上“其他”、“开心”、“悲伤”、“愤怒”四个类别的样本数量分布约为 3:1:1:1，而在验证集和测试集上四个类别的样本数量分布约为 22:1:1:1 。可见在验证集和测试集上“其他”一类的样本数量要远高于其他三个类别，和训练集相比其样本占比也相对较高，而另外三个类别的样本数量在各个数据集上则大致相同。

在比赛的最终测试阶段，训练集和验证集均已公布情感标注并且被允许用于模型训练，因此我们结合了原本的训练集和验证集作为我们的训练数据。

4.3.1　文本长度

我们对数据集的文本进行分词后统计了各类别样本的单词数量分布，以下简称为文本长度。表4.1显示在训练集上各情感类别的样本在每轮发言的文本长度分

表 4.2　各情感类别对应的三轮对话例子

类别	对话
开心	（第一轮）用户甲: live in uttra khand （第二轮）用户乙: ohh nice! love that place! （第三轮）用户甲: 😂😂
悲伤	（第一轮）用户甲: Not coz of you （第二轮）用户乙: why? Tell me （第三轮）用户甲: :(My girlfriend left me
愤怒	（第一轮）用户甲: He is over me （第二轮）用户乙: so YOU say （第三轮）用户甲: I just hate him
其他	（第一轮）用户甲: degreee （第二轮）用户乙: what degree & where? （第三轮）用户甲: sryyy i really got to goo

布，可以看出在每轮发言中，不同类别样本的文本长度分布大致相同。虽然第一轮和第三轮的样本中最长的文本分别达到 146 个词和 74 个词，但各轮样本的文本长度大部分在 22 个词以内，约为前一章实验数据中样本文本长度的一半。

4.3.2　文本特征

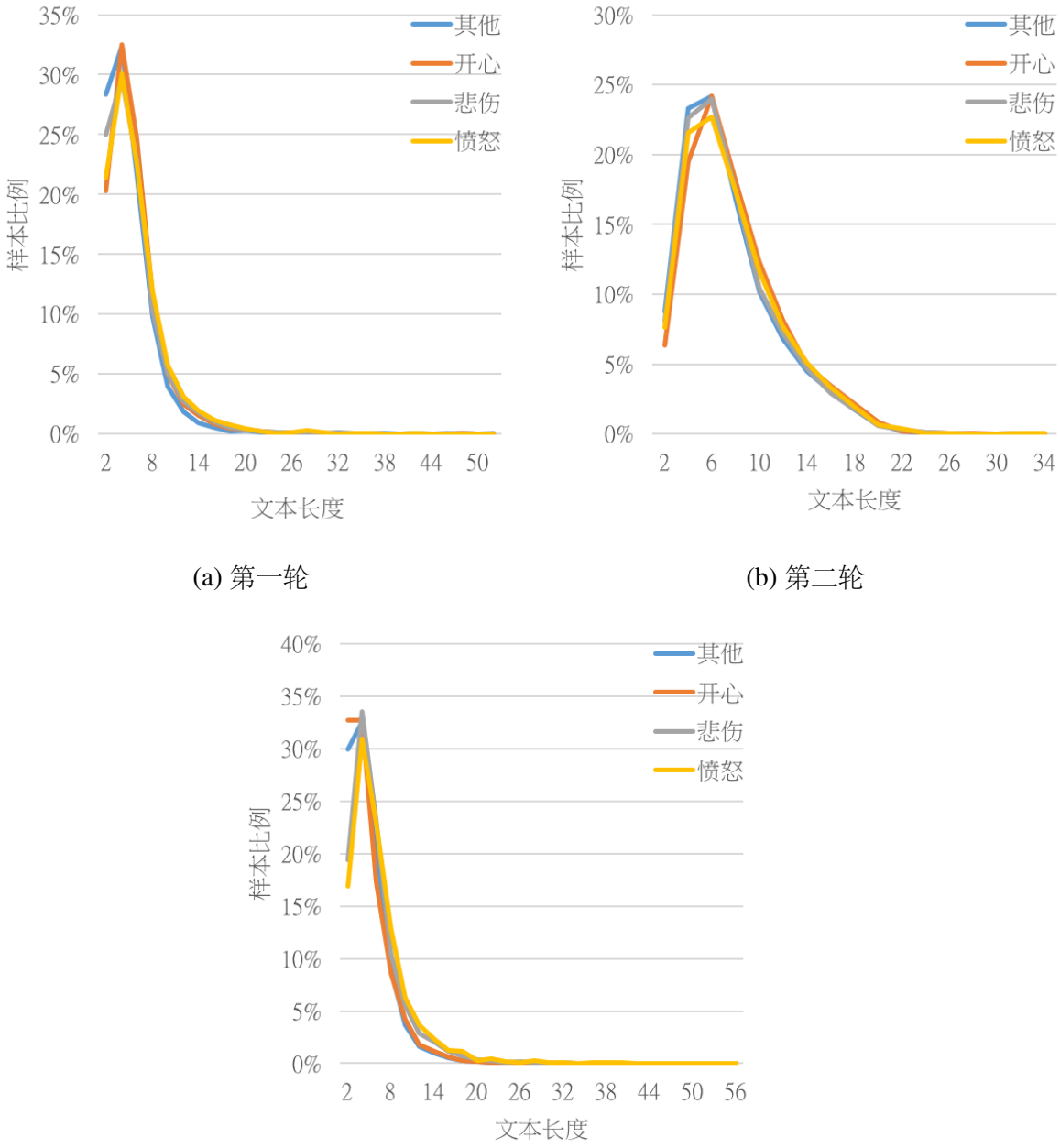
对于比赛中提供的数据集，比赛组织者并没有给出数据的具体来源，但经过人工观察后我们可以发现一些社交网络上常见的文本特征，其中出现频率较高的特征如下：

- 大量的缩略词使用，如“u”代替“you”, “y”代替“why”, “im”代替“I am”。
- 出现在句子最前或最后的表情符，其中包括 Unicode 定义表情符（如😂）和由标点符号组成的表情符（如“:-D”）两类。
- 一些在社交媒体平台上常见的、有别于正规英语的用法，如拼写错误、全大写字母的单词等，可以参考章节2.4.1描述的例子。

4.3.3　各轮发言间的情感信息

为了结合三轮对话中前两轮的发言来预测最后一轮发言的情感，我们需要观察各轮发言是否关最后一轮发言的情感有关，若有关的话各轮发言是否起着相同的作用。以下我们将采用数据集中的具体例子说明我们对语料的观察和理解。

首先，对于三轮发言中两个用户的情感，我们发现用户甲和用户乙表达的情感可能截然不同，参考以下例子：



(a) 第一轮　　(b) 第二轮

(c) 第三轮

图 4.1　训练集上各情感类别的样本在每轮发言的文本长度分布

（第一轮）用户甲:　yes yay fun

（第二轮）用户乙:　are not you joining us? :(

（第三轮）用户甲:　yes

用户甲在最后一轮发言的情感标签为“开心”，而第一轮发言在字面意思上同样偏向正面。而对于用户乙在第二轮的发言，根据上下文意思和最后文本末尾的“:(”，我们可以推断用户乙的情感为“伤心”。可见三轮对话中两个用户表达的情感可能不同甚至矛盾，因此用户甲的发言和用户乙的发言对识别第三轮发言的情感应起着

不同的作用，第二轮中用户乙的发言甚至没有提示的作用。

另外对于用户甲，我们发现他在第一轮和第三轮中表达的情感也有可能不同，参考以下例子：

（第一轮）用户甲：　not fine

（第二轮）用户乙：　why? :'o

（第三轮）用户甲：　tomorrow lab exam

用户甲在最后一轮发言的情感标签为“悲伤”，而在字面意思上，最后一轮发言的情感偏中性，只是在陈述“明天有考试”的事情，“悲伤”的情感提示主要来自用户甲在第一轮中表示自己的情况“不太好”，第三轮发言在解释“不太好”的原因。基于此例子可以认为，同为用户甲发言的第一轮在某此情况下对识别第三轮发言的情感起着决定性作用。再观察下面的例子：

（第一轮）用户甲：　ohh sorry

（第二轮）用户乙：　do not worry, you are not the first

（第三轮）用户甲：　glad to hear

用户甲在最后一轮发言的情感标签为“开心”，对于用户甲在第一轮发言表达的情感，结合用户乙在第二轮的发言，可以认为用户甲在第一轮发言中表示对用户乙的同情，这与“开心”为不同的情感，而在用户乙发言后，用户甲的情感因为用户乙说的话而转换成“开心”。可见用户甲在第一轮和第三轮中表达的情感有可能不同，这种情感的转变可能由用户乙的发言造成，但无论具体原因是什么，用户甲在第一轮和第三轮中的发言对识别第三轮发言的情感应起着不同的作用。

再者，我们发现用户甲在第三轮发言中可能出现多于一种情感，参考以下例子：

（第一轮）用户甲：　😃 yes yes

（第二轮）用户乙：　:3 you seem like a happy person

（第三轮）用户甲：　yes 😂 happy outside, 😞 sad inside

用户甲在最后一轮发言的情感标签为“悲伤”，而在文本上，第三轮发言的前半表现为正面情感，后半表现为负表情感，前后的情感不同。对于当第三轮发言中出现两种情感时情感标签以何者为准，比赛组织者并没有给出细节的说明，但对语料的观察可以发现普遍以后面出现的情感为主。

最后总结我们对语料的观察，我们发现三轮发言对识别第三轮发言的情感应起着不同的作用。当第三轮发言有明显的情感表达，我们可以无视前两轮的发言得出识别结果。当第三轮发言表达的情感较隐晦或偏中性，需要参考同为用户甲发言的第一轮的信息。第二轮发言对第三轮发言的情感识别未有明显的提示作用。

4.4　框架设计

同样地，我们提出了一个基于多步决策的情感识别系统。对于原本的情感四分类问题，我们把它拆解成了以下三个子分类问题的叠加：

1. 原本的情感四分类问题。
2. “开心”、“悲伤”和“愤怒”三分类。
3. “其他”和“不是其他”二分类。

对于每个子分类问题，我们分别准备其对应的分类器组，依次命名为第一、第二、第三组分类器，分别由 N_1、N_2、N_3 个分类器组成。那么对于三轮对话中最后一轮发言的情感识别，我们系统的决策过程如下：

- 首先由第一组分类器进行多数投票，得出预测标签 $Label^1_{MV}$，直接以它作为第一步的预测标签 $Label_I$ 。以下把系统到这一步为止的判断结果称为中间结果 I。
- 第二步，由第二组分类器进行多数投票，得出预测标签 $Label^2_{MV}$，若超过 thr_2 分类器投票投给 $Label^2_{MV}$ 且第一步的预测标签 $Label_I$ 为“其他”以外的三种情感类别之一，则把预测结果修改为 $Label^2_{MV}$，否则保持不变，以此得出第二步的预测标签 $Label_{II}$。以下把系统到这一步为止的判断结果称为中间结果 II。
- 最后一步，由第三组分类器给出投票结果，若超过 thr_3 个分类器投票给“其他”且第二轮的预测标签 $Label_{II}$ 不是“其他”，则把预测标签修改为“其他”，否则保持不变，以此得出第三步的预测标签 $Label_{III}$，同时作为整个系统对第三轮发言最终的情感识别结果。

整个决策过程可以分成两大部分。第一部分的目的是初步完成对第三轮发言的四分类情感识别，对应上述三步决策中的第一步。第二部分的目的是基于第一部分的初步识别结果逐步进行修正，每一步只关注一个子分类问题，对应上述三步决策中的后两步。其中第二步进行“开心”、“悲伤”和“愤怒”的三分类，目的在于调整三个类别之间的识别结果。而在最后一步进行“其他”和“不是其他”二分类，但特别地不是取多数投票，而是只要超过 thr_3 个分类器投票给“其他”即把识别标签改成“其他”，目的在于提高系统对“其他”一类的召回率。

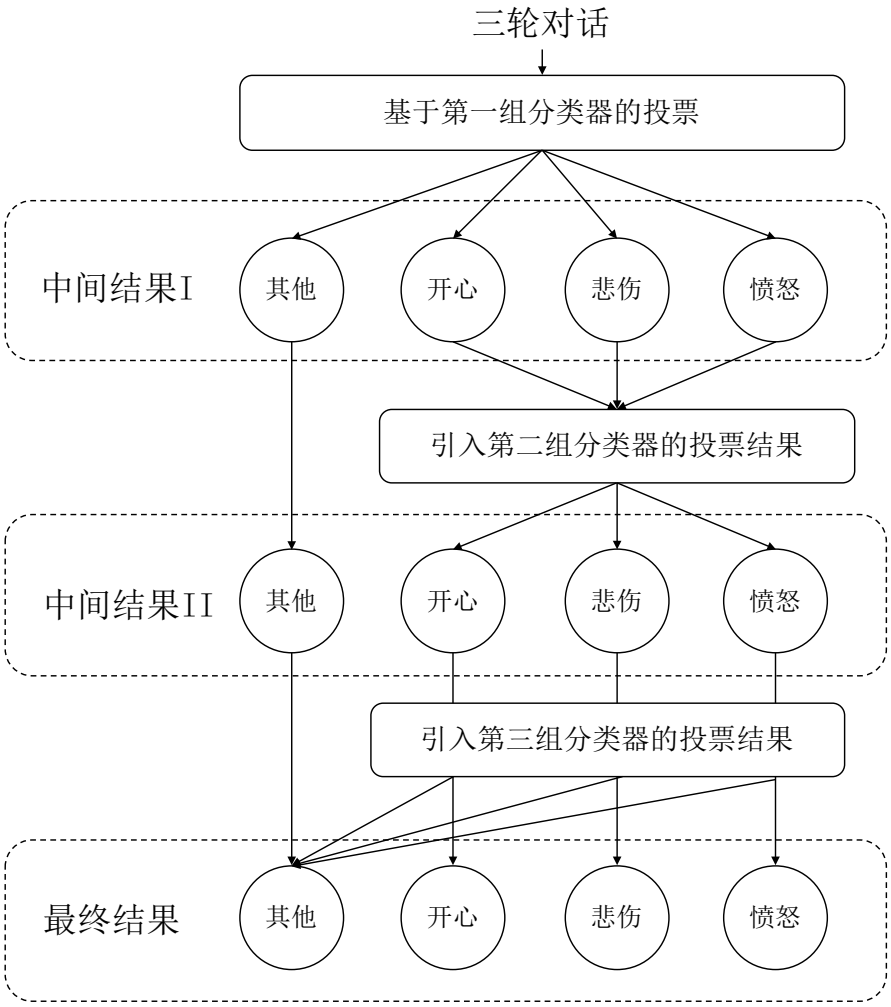


图 4.2　三轮对话的情感识别系统框架

其中对于每个子分类问题，我们都采用了相同的模型框架，如图4.3所示，每个子分类器的输入为三轮对话对应的词序列 $\{w^1, w^2, w^3\}$。根据在章节4.3.3中的分析，我们认为三轮发言对识别第三轮发言的情感识别起着不同的作用，因此三轮发言对应的词序列分别进入不同的通道。每个通道的结构相同，第一步都是把词序列转换成对应的词嵌入向量序列，作为特征编码器的输入。此处特征编码器的设计与章节3.4中的相同，目的是把单轮发言对应的词向量序列转换成固定长度的特征向量，以此得出该轮发言的特征向量。为了简化，三个通道的特征编码器采用相同的模型，但考虑到每轮发言对第三轮情感有关的特征可能不同，三个通道的模型各自采用不同的权重。分别得出三轮发言的特征向量后我们需要结合这些特征来识别第三轮的情感类别，此处我们直接把三个向量拼接成一个向量，作为概率预测器的输入。此处概率预测器的设计与章节3.4中的相似，但实现上采用了两层的全联接层，第一层的激活函数为线性整流函数（Rectified Linear Unit, ReLU），

而第二层的激活函数依然为 $Softmax$，以得出各个情感类别的概率分布。

考虑到对于不同情感类别的语言特征不同，各个模型的建模能力也会有所不同。所以对于每个子分类问题，我们会分别比较各个模型的性能，以求在子分类问题上达到尽可能好的识别性能。

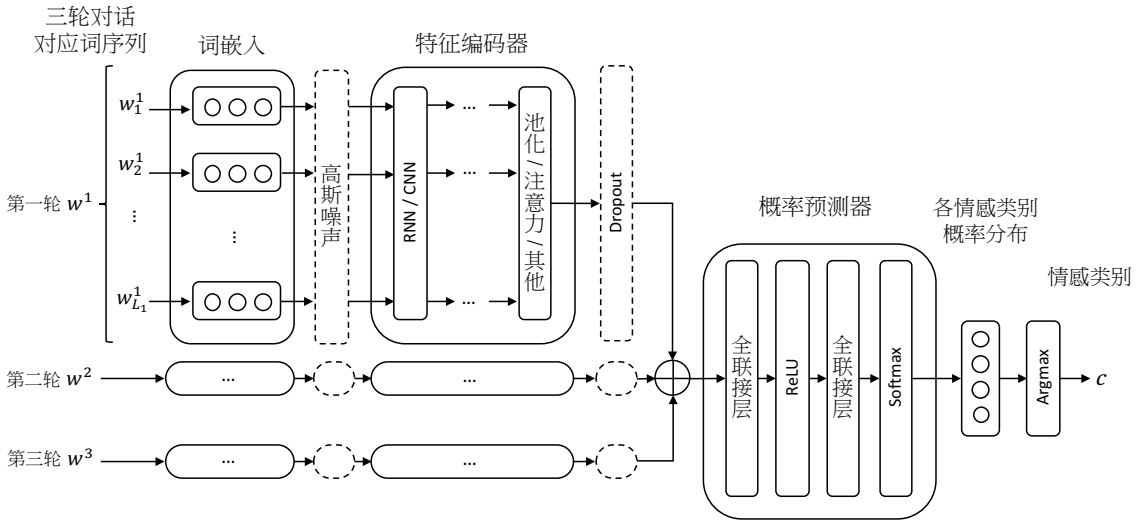


图 4.3　三轮对话的情感识别子分类器模型框架

4.5　实验与分析

4.5.1　数据预处理

基于我们在章节4.3.2中对样本文本的观察，我们依次采取了以下数据预处理手法

- 对于全字母大写的文本段，在该段文本的前后添加“<allcap>”和“</allcap>”示意。如“YAYYYY”替换成序列“<allcap>”、“yayyyy”、“</allcap>”。
- 对于重复次数大于等次三次的标点符号，以“<repeated>”示意，如“!!!”替换成序列“!”、“<repeated>”，表示“!”被多次重复，即时假设重复的具体次数和原微博的反讽类别无关。
- 对于字母被故意重复的单词，以“<elongated>”示意，如“Noooooooo”替换成序列“No”、“<elongated>”，表示“No”中某一个或多个字母被多次重复，即假设被重复的字符和具体重复的次数和原微博的反讽类别无关。
- 将数字串替换成“<num>”，将电话号码替换成“<phone>”，将日期和时间分别替换成“<date>”和“<time>”，将数字百分比替换成“<percentage>”，超链接替换成“<url>”，即假设其中的具体数值和原微博的反讽类别无关。
- 对于由多个标点符号组成的表情符，我们将其替换成对应的情感标签，如将

“:)”替换成“<happy>”, “:((”替换成“<sad>”。

- 在完成以上处理后，对英文的大小写统一转换成小写。

具体实现和章节 3.5.1中的相同。

4.5.2　评价指标

按照国际比赛 SemEval-2019 任务三的设置，系统性能以针对“开心”、“悲伤”和“愤怒”三个情感类别的 F1 值（以下记为 F_μ）作为识别系统性能的主要评价指标。其定义如下：

$$F_\mu = \frac{2 \times P_\mu \times R_\mu}{P_\mu + R_\mu} \tag{4-1}$$

其中 P_μ 和 R_μ 分别是针对“开心”、“悲伤”和“愤怒”三个情感类别的正确率和召回率，其定义如下：

$$P_\mu = \frac{\sum\limits_{c \in C} TP_c}{\sum\limits_{c \in C} (TP_c + FP_c)} \tag{4-2}$$

$$R_\mu = \frac{\sum\limits_{c \in C} TP_c}{\sum\limits_{c \in C} (TP_c + FN_c)} \tag{4-3}$$

其中 C 对应情感类别集合 {开心，悲伤，愤怒}，TP_c 表示被系统预测为类别 c，且真实标签为类别 c 的样本数量；FP_c 表示被系统预测为类别 c，但真实标签不是类别 c 的样本数量；FN_c 被系统预测为不是类别 c，但真实标签为类别 c 的样本数量。

4.5.3　模型训练

对于每个子分类问题，我们分别从完整的训练数据和测试数据中筛选出对应类别的样本作为实验数据。并采用相同的方法进行模型训练，其中涉及一些策略如下：

- 对于涉及类别“其他”的子分类问题，我们在模型训练前会基于“其他”的每一个样本随机生成一个新的临时样本。生成方法是在三轮发言中随机选择一轮，再在该轮发言的文本中随机选择一个词删掉，以此模拟用户使用未被识

别的单词或错拼词，得出一个“其他”的新样本。这使得模型对识别“其他”有更好的鲁棒性，但这会同时导致模型倾向于将更多不是“其他”的样本识别为“其他”。对于为什么只基于类别“其他”的样本生成新的临时样本，原因在于语料中类别为“其他”的三轮对话在忽略一个单词后并不会带来情感类别的变化。但对于类别“开心”、“悲伤”和“愤怒”，大部分样本仅以一个或几个单词表达其情感，若其中一个单词被删除会直接导致其情感表达的变化，为避免产生有误导性的临时样本，我们只基于类别“其他”的样本生成新的样本。

- 对于训练数据，我们分别从各个类别的样本中随机选出 90% 的样本作为训练集，剩余 10% 的样本作为验证集，以保留各个类别的样本分布。在每一轮模型参数调整后，计算各组模型参数在验证集上的某个指标，经过有限轮迭代后，取在验证集上性能指标最优的网络参数作为该分类器的参数，以此避免在训练数据上过拟合。对第一组和第二组分类器，我们以“开心”、“悲伤”和“愤怒”的正确率 P_μ 作为选择参数的指标，原因在于测试集中“开心”、“悲伤”和“愤怒”的样本占比要低于验证集（参考章节 4.3），这导致模型在测试集上会明显把较多“其他”的样本误判为这三种情感类别之一，正确率 P_μ 的数值会明显下降，间接拉低了 F1 值 F_μ，故选择正确率 P_μ 来确保模型的正确率 P_μ 尽可能高。对于第三组分类器（“其他”和“不是其他”二分类），我们采用针对类别“其他”的正确率作为选择参数的性能指标，目的在于召回“其他”样本的同时尽可能保证其正确率。
- 由于在我们的系统中，每个子分类问题由多个分类器经过投票给出识别结果，为了充分运用训练数据，在训练过程中每个分类器的验证集为独立随机筛选得出，一方面使得每个训练样本都有概率被用于某个分类器的模型训练，另一方面各个分类器的训练集不同，避免了对部分数据的过拟合。
- 对于词嵌入层，我们利用了章节2.4.2.1中 Baziotis 等人[47] 提供的词嵌入模型，直接用于初始化词嵌入层的参数。而对于词嵌入模型中未被覆盖的单词，若它在至少 2 个训练样本中出现，则为其随机生成词向量。在训练过程中，我们不对词嵌入层的参数进行调整。
- 在参数训练阶段，我们对词嵌入层输出的词向量序列添加高斯噪音。由于词嵌入算法在原理上使得意思相似的单词投影到词嵌入空间中距离相近的点上，高斯噪音的添加相当于把原本的单词替换成近义词，使得模型能更好地理解近义词构成的语言模式，另一方面缓解过拟合的问题。在验证阶段和测试阶段，高斯躁音的标准方差被调整为零，即不起作用。
- 在参数训练阶段，我们在特征编码器和概率预测器之间添加了 Dropout 层，

以概率 $p_{Dropout}$ 将特征编码器得出特征向量上的各位数值置为零，并对没有被置零的各位数值乘以常量 $\frac{1}{1-p_{dropout}}$。在验证阶段和测试阶段，Dropout 层不起作用。

- 对于模型训练的损失函数，我们以权重 l_2 加入了概率预测器中两个全联接层的权重（不包括偏移量）的 L2 正则项。

4.5.4　实验结果与分析

以下实验主要分成两大部分。第一部分是分析不同模型在不同子分类问题下的性能，根据章节4.4，我们最终的反讽识别框架涉及以下多个子分类问题：区分“开心”、“悲伤”、“愤怒”和“其他”的四分类问题，区分“开心”、“悲伤”和“愤怒”的三分类问题、区分“其他”和“不是其他”的二分类问题。对于各个子分类问题，我们基于章节 4.4中的子分类模型框架进行实验，透过采用不同配置了解不同模型对三轮对话的情感识别能力。

第二部分对我们提出的基于多步决策的的识别系统进行研究。一方面评估我们系统的整体性能水平，另一方面透过观察系统的中间结果分析多步决策如何对系统的整体性能带来影响，从而解释此系统设计的合理性。

4.5.4.1　面向情感四分类的模型性能分析

对于面向三轮对话的情感四分类问题，表 4.3和图 4.4显示各个模型在测试集上的性能。注意表中的正确率、召回率、F1 值为针对“开心”、“悲伤”和“愤怒”的指标，对应比赛 SemEval2019 任务三关注的主要指标，即章节 4.5.2中定义的 P_μ、R_μ 和 F_μ。

对于 F1 值，CNN 达到最好的 0.7437 ，其次是的 BiGRU 配合注意力机制和 CNN 配合注意力机制，两者和第一的 CNN 差距较大（约 0.0384）。对于准确率，同样是 CNN 的数值最高（0.9245），第二和第三分别是 BiGRU 配合注意力机制和 GRU 配合注意力机制，同样地和 CNN 差距较大（约 0.0102）。对于正确率，则是 GRU 配合注意力机制的性能最优（0.7330），其次的 GRU 和 CNN 和前者差距较小（仅 0.0034）。对于召回率，第一位依然是 CNN（0.7584），其次是 CNN 配合注意力机制，和前者差距约 0.0156，第三的 BiGRU 配合注意力机制则和前两者差距较大（数值低于 0.7）。

总的来说，CNN 同时在 F1 值、准确率和召回率三项指标上达到了最好的性能，而正确率同样逼近最好的 GRU 配合注意力机制，可见 CNN 在此子分类问题上有显着较好的建模能力。另外值得注意的是 LSTM 配合注意力机制在各项指标

中数据最低，其中正确率、召回率和 F1 值都远低于其他模型，经过检查后发现该模型在多次训练中都会忽略“开心”、“悲伤”和“愤怒”中的某个类。在排除算法实现有问题的可能性后，我们认为这是由于模型无法同时对四个类别的数据建模所致。

表 4.3　面向情感四分类器的模型性能

	准确率	正确率	召回率	F1 值
CNN	**0.9245 (1)**	0.7295 (3)	**0.7584 (1)**	**0.7437 (1)**
CNN+ 注意力机制	0.9071 (5)	0.6574 (8)	0.7428 (2)	0.6975 (3)
GRU	0.9052 (8)	0.7296 (2)	0.5481 (9)	0.6259 (9)
GRU+ 注意力机制	0.9105 (3)	**0.7330 (1)**	0.5974 (8)	0.6583 (7)
BiGRU	0.9067 (7)	0.6598 (7)	0.6923 (4)	0.6757 (5)
BiGRU+ 注意力机制	0.9143 (2)	0.7162 (4)	0.6947 (3)	0.7053 (2)
LSTM	0.9071 (5)	0.6709 (6)	0.6911 (5)	0.6809 (4)
LSTM+ 注意力机制	0.8553 (10)	0.2994 (10)	0.2788 (10)	0.2887 (10)
BiLSTM	0.8969 (9)	0.6331 (9)	0.6659 (6)	0.6491 (8)
BiLSTM+ 注意力机制	0.9092 (4)	0.7067 (5)	0.6226 (7)	0.6620 (6)

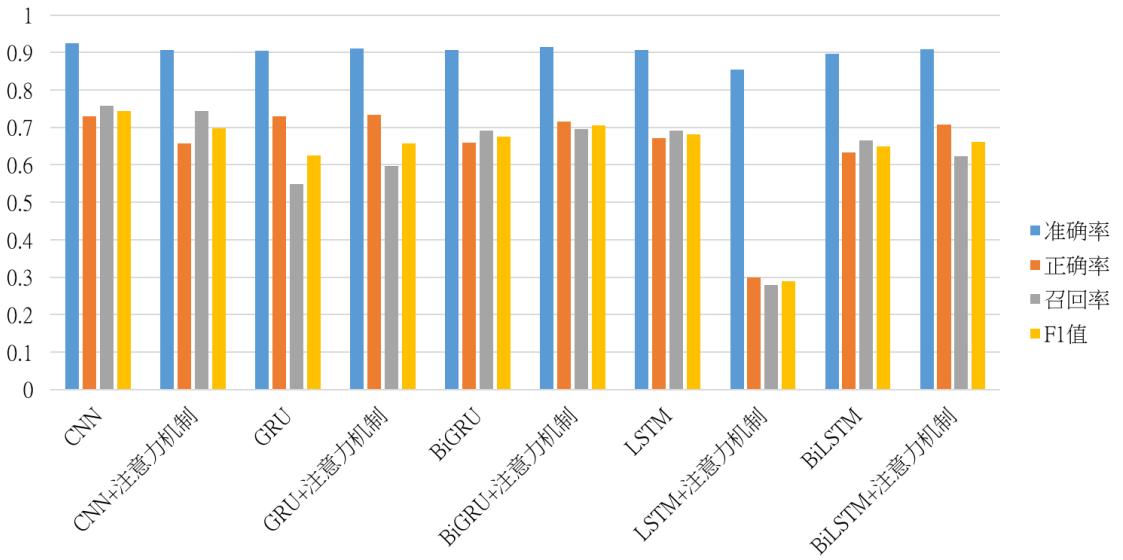


图 4.4　面向情感四分类器的模型性能

4.5.4.2　面向“开心”、“悲伤”和“愤怒”三分类的模型性能分析

对于面向三轮对话的“开心”、“悲伤”和“愤怒”三分类问题，表 4.4和图 4.5显示各个模型在测试集上的性能。注意表中的正确率、召回率、F1 值同样对应章节 4.5.2中定义的 P_μ、R_μ 和 F_μ。

对于表中的四个指标，均由 CNN 达到最好的效果。对于 F1 值，排第二和第三的分别是 CNN 配合注意力机制以及 GRU 配合注意力机制，两者和 CNN 的 F1 值差距明显（约 0.0179）。对于准确率，第二的 GRU 配合注意力机制和第三的 CNN 配合注意力机制的数值接近，但都和 CNN 有一定差距（约 0.132）。对于正确率，第二为 CNN 配合注意力机制，数值明显低于 CNN（差距约 0.0102）。对于召回率，由 GRU 配合注意力机制、BiGRU 配合注意力机制、BiLSTM 配合注意力机制并排第二，与 CNN 的召回率差距明显（约 0.0219）。

另外，虽然添加注意力机制对 CNN 的性能依然没有正面作用，但 GRU、BiGRU 和 BiLSTM 在添加注意力机制后都有较明显的提升，显示注意力模型有效捕捉此问题背后的某种语言特征。

表 4.4　面向“开心”、“悲伤”和“愤怒”的三分类器模型性能

	准确率	正确率	召回率	F1 值
CNN	**0.9507 (1)**	**0.9454 (1)**	**0.9471 (1)**	**0.9462 (1)**
CNN+ 注意力机制	0.9351 (3)	0.9352 (2)	0.9215 (5)	0.9283 (2)
GRU	0.9315 (6)	0.9278 (4)	0.9142 (9)	0.9210 (6)
GRU+ 注意力机制	0.9375 (2)	0.9303 (3)	0.9252 (2)	0.9277 (3)
BiGRU	0.9303 (7)	0.9212 (8)	0.9179 (7)	0.9196 (8)
BiGRU+ 注意力机制	0.9339 (4)	0.9235 (6)	0.9252 (2)	0.9243 (5)
LSTM	0.9291 (8)	0.9229 (7)	0.9179 (7)	0.9204 (7)
LSTM+ 注意力机制	0.9279 (9)	0.9180 (9)	0.9197 (6)	0.9189 (9)
BiLSTM	0.9279 (9)	0.9176 (10)	0.9142 (9)	0.9159 (10)
BiLSTM+ 注意力机制	0.9339 (4)	0.9269 (5)	0.9252 (2)	0.9260 (4)

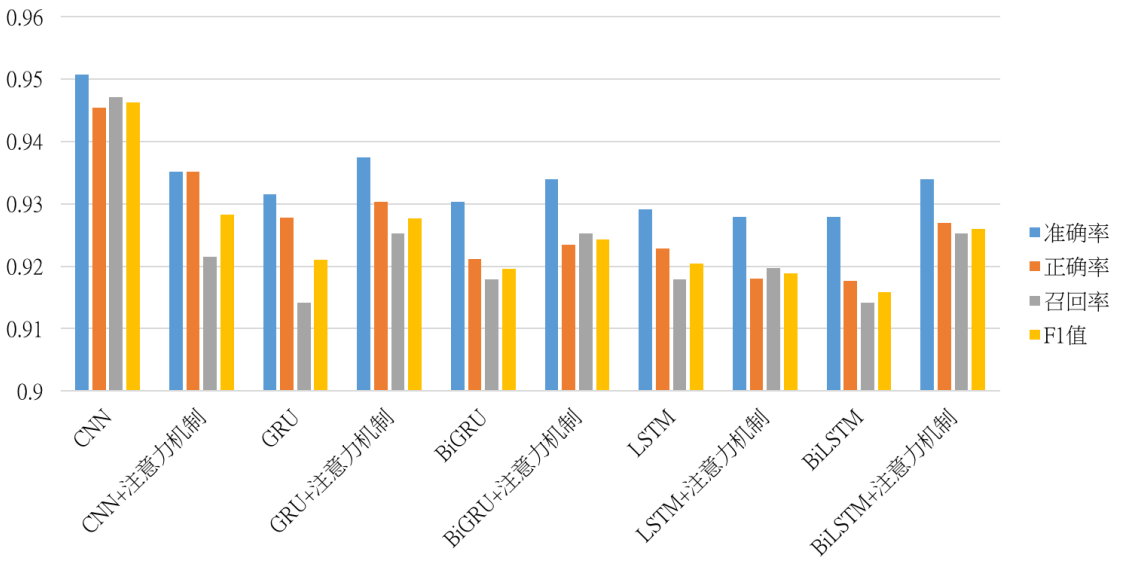


图 4.5　面向“开心”、“悲伤”和“愤怒”的三分类器模型性能

4.5.4.3　面向“其他”和“不是其他”二分类的模型性能分析

对于面向三轮对话的“其他”和“不是其他”二分类问题，表 4.5和图 4.6显示各个模型在测试集上的性能。注意表中的正确率、召回率、F1 值同样对应“其他”一类的指标。

对于 F1 值，GRU 配合注意力机制达到最高数值的 0.9546, 其次的 CNN 和 BiGRU 配合注意力机制都和前者的数值接近（分别为 0.9540 和 0.9494）。对于准确率，排名前三的模型同样依次是 GRU 配合注意力机制、CNN 和 BiGRU 配合注意力机制，前两者数值比较接近（0.9230 和 0.9227），第三的数值（0.9152）则和前两者差距较明显。对于正确率，由 GRU 达到最好的性能（0.9670），其次的 LSTM（0.9649）和 CNN（0.9634）的性能也比较接近。对于召回率，排名第一第二的依然是 GRU 配合注意力机制（0.9549）和 CNN（0.9448），CNN 配合注意力机制以及 LSTM 配合注意力机制并排第三（0.9391）。

虽然 GRU 配合注意力机制在三项指标上都排名第一，但其正确率却是各个模型中排名倒数第一。相对地，对正确率排名第一第二的 LSTM 和 GRU，它们在另外三项指标上却都排名倒数第一第二。可见数据不均匀会影响模型的识别倾向，甚至出现模型只在个别指标上明显靠前但同时也其他指标上明显靠后的情况。

表 4.5　面向“其他”和“不是其他”的二分类模型性能

	准确率	正确率	召回率	F1 值
CNN	0.9227 (2)	0.9634 (3)	0.9448 (2)	0.9540 (2)
CNN+ 注意力机制	0.9116 (7)	0.9560 (9)	0.9391 (3)	0.9475 (7)
GRU	0.9074 (9)	**0.9670 (1)**	0.9224 (9)	0.9442 (9)
GRU+ 注意力机制	**0.9230 (1)**	0.9553 (10)	**0.9540 (1)**	**0.9546 (1)**
BiGRU	0.9134 (4)	0.9630 (4)	0.9339 (6)	0.9482 (4)
BiGRU+ 注意力机制	0.9152 (3)	0.9624 (6)	0.9367 (5)	0.9494 (3)
LSTM	0.9011 (10)	0.9649 (2)	0.9168 (10)	0.9402 (10)
LSTM+ 注意力机制	0.9121 (6)	0.9567 (8)	0.9391 (3)	0.9478 (6)
BiLSTM	0.9089 (8)	0.9595 (7)	0.9320 (8)	0.9456 (8)
BiLSTM+ 注意力机制	0.9131 (5)	0.9627 (5)	0.9337 (7)	0.9480 (5)

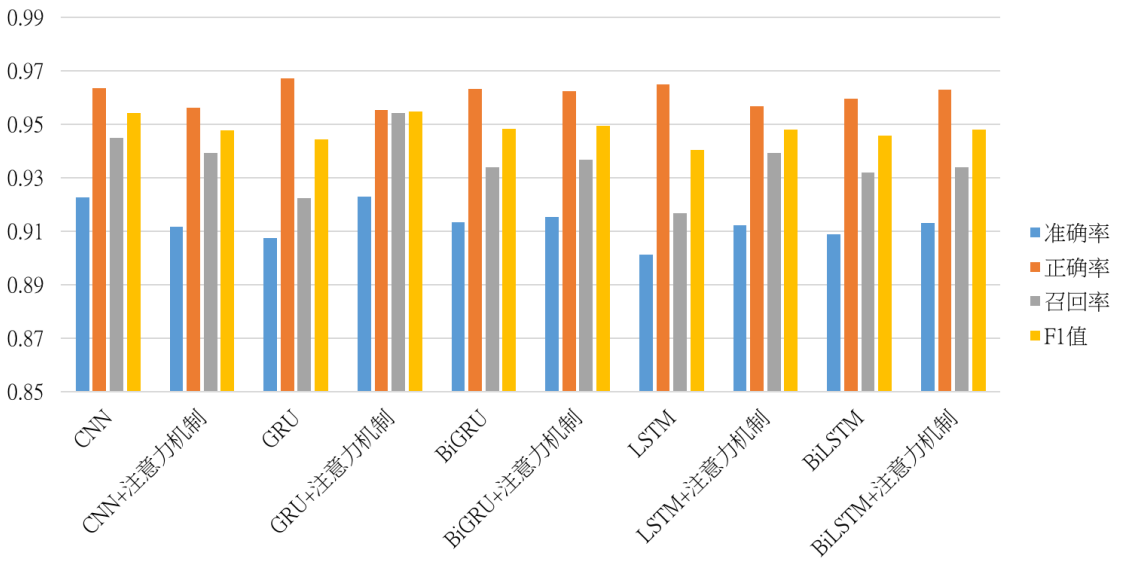


图 4.6　面向“其他”和“不是其他”的二分类模型性能

4.5.4.4　面向三轮对话的情感识别系统的性能分析

经过前面各章节的分析，我们已经对各个子分类问题中不同模型的性能有大致了解，接下来我们将研究由多组子分类器集成得出的情感识别系统的性能，并和比赛 SemEval-2019 任务三的参赛系统比较来评估我们系统的性能水平。

按照章节 4.4，我们的情感识别系统涉及三个子分类问题。对于和原问题相同的情感四分类问题，根据章节 4.5.4.1的实验结果，我们采用 CNN 作为第一组分类器的模型。对于面向“开心”、“悲伤”和“愤怒”的三分类问题，按照章节 4.5.4.2的实验结果，我们同样选用 CNN 作为第二组分类器的模型。对于面向“其他”和“不是其他”的二分类问题，按照章节 4.5.4.3的实验结果，我们采用 GRU 作为第三组分类

器的模型。按照章节 4.5.3，我们基于选定的模型分别为每个分类器组训练 5 个分类器。

最后按照章节 4.4中给出的识别流程，透过逐步结合各组分类器的投票结果得出三轮对话中最后一轮所表达的情感类别。对于系统中对第二、三组分类器需要配置的参数 thr_2 和 thr_3，我们均设置为 3。

表 4.6显示我们的系统和 SemEval-2019 任务三中其他参赛系统在测试集上最后一次提交的性能。注意表中的 F1 值为针对“开心”、“悲伤”和“愤怒”的指标，即章节 4.5.2中定义的 F_μ。在 SemEval-2019 任务三的最终测试阶段，参赛系统共 165 个，但官方只给出了各个参赛系统的 F1 值。表中仅显示排名前 10（按照 F1 值排名）的参赛系统，其中队伍名称为 THU_HCSI 对应我们当时提交的系统，排名第 8。

表 4.6　SemEval-2019 任务三前十名参赛系统性能

排名	队伍名称	F1 值
1	PingAn GammaLab	0.7959
2	(无)	0.7947
3	NELEC	0.7765
4	SymantoResearch	0.7731
5	ANA	0.7709
6	CAiRE_HKUST	0.7677
7	SNU_IDS	0.7661
8	**THU_HCSI**	0.7616
9	(无)	0.7608
10	YUN-HPCC	0.7588

另外，由于我们的系统在每步决策后都可以作为一组预测结果，因此我们观察了每个中间结果在测试集上的性能。以下的系统由重新训练的子分类器组成，由于子分类器的训练存在随机性（参考章节 4.5.3），因此和我们当时的参赛系统的 F1 值不同，但并不影响我们分析系统中每步决策的作用。

表 4.7中第一行“中间结果 I”对应直接由一组情感四分类器（第一组分类器）进行多数投票的识别结果。第二行“中间结果 II”是基于“开心”、“悲伤”和“愤怒”三分类器组（第二组分类器）的投票对原本被识别为这三个类别之一的样本重新进行分类的识别结果，可见四项指标都有所提升。再对比中间结果 I 和中间结果 II 的混淆矩阵（分别对应表 4.8和表 4.9）可以发现 4 个样本的识别结果被正确修正，2 个“愤怒”的样本被误判为“悲伤”，这显示了即使引入阈值 thr_2 来评估投票的可信度，

由子分类器之间投票得出的识别结果依然会存在判断出错的情况，但整体上还是加强了对“开心”、“悲伤”和“愤怒”的区分能力。

表中最底一行的“最终结果”是基于“其他”和“不是其他”的二分类器组（第三组分类器）的投票结果尝试召回更多“其他”的样本，可见除了召回率有稍微下降，其他三项指标都有所提升，虽然准确率只稍微上升，但正确率的明显提升大大提高了最终的 F1 值。再对比中间结果 II 和最终结果的混淆矩阵（分别对应表 4.9和表 4.10）可以发现 11 个“其他”的样本被正确召回，间接提高了“开心”、“悲伤”和“愤怒”的正确率 P_μ，但同时误判了 1 个原本正确的样本，故召回率 R_μ 下降。同样地第三组分类器在引入阈值 thr_3 后依然会为识别结果带来错误的修改，但被正确召回的样本比错误的多，整体上对系统的识别结果也是起着正面的作用。

接下来，我们分析一下选择面向“开心”、“悲伤”和“愤怒”三分类以及面向“其他”和“不是其他”二分类这两个子分类问题的合理性。首先对于面向“开心”、“悲伤”和“愤怒”的三分类问题，参考章节4.5.4.2可以发现我们的模型在该子问题上有很好的区分能力（F1 值达到 0.9462），其数值明显高于原本的四分类问题（对比章节4.5.4.1中最高的 F1 值为 0.7437），可见在引入类别“其他”后对 F1 值有较大影响，一方面是数据量的增加，另一方面说明模型同时区分“其他”和另外三个类别存在较明显的困难。另外，回顾章节4.5.4.1、4.5.4.2和4.5.4.3可以发现前两个子问题中 F1 值最高的都是 CNN，而后者 F1 值最高的是 GRU 配合注意力机制，显示“其他”和另外三个类别背后的数学模型应该稍有不同。因此在第二步中我们以 CNN 重新对被识别为“开心”、“悲伤”和“愤怒”的样本进行区分，然后在第三步中以不同的模型来识别出原本没有被召回的“其他”的样本，以结合两种模型的识别能力。

对于为什么只召回“其他”的样本，而没有反过来召回“不是其他”的样本再进行三分类，这和各类别样本量的分布有关。“其他”的样本量在测试上明显多于另外三个类别的样本量，如果尝试进一步召回“不是其他”的样本，系统会同时把更多“其他”的样本误判为“不是其他”，这就导致正确率 P_μ 的下降显着大于召回率 R_μ 的提升，从而导致核心指标 F_μ 下降。而相反地从原本被识别为“开心”、“悲伤”和“愤怒”的样本中召回“其他”的样本，虽然会把部分原本识别正确的样本误判为“其他”，但会有较大概率召回更多正确的“其他”的样本，间接提高正确率 P_μ，因此我们选择召回更多“其他”的样本作为第三步的核心目标。

4.5.5　错误分析

本节中，我们将对系统错误识别的样本进行观察，分析其中的原因，从而对问题有更探入的了解，以及尝试给出下一步的改进方向。以下使用的例子均参照

表 4.7　我们的多步决策识别系统的中间结果和最终结果在测试集上的性能

	准确率	正确率	召回率	F1 值
中间结果 I	0.9278	0.7360	0.7740	0.7545
中间结果 II	0.9281	0.7383	**0.7764**	0.7569
最终结果	**0.9299**	**0.7474**	0.7752	**0.7611**

表 4.8　测试集上中间结果 I 对应的混淆矩阵

		预测标签			
		其他	开心	悲伤	愤怒
真实标签	其他	4467	65	57	88
	开心	82	196	4	2
	悲伤	38	2	201	9
	愤怒	47	1	3	247

表 4.11。

- **词组情感倾向识别**。参考例子 1，第一轮中用户甲的“in mood”表示心情愉快，属于正向情感，第三轮中的“yeah”也略带正向情感，按照章节 4.3.3，应该得出样本情感标签为“开心”。然而“in mood”在训练集中主要以“not in mood”出现，且这些样本标签都是“悲伤”，显然系统未能正确识别“in mood”本身的情感倾向导致了识别错误，另外可以推测对于在训练集上没有出现过的词组也会有相同的问题。在章节 3.5.5中相同，只能透过额外引入语言知识解决。
- **对于同时出现多种情感的处理**。参考例子 2，第三轮中用户甲的发言在字面上包含两种情感，首先是“Haha”表达偏向“开心”的情感，“Bite me!😒”则带有鄙视和不满的意思，倾向于“悲伤”。按照章节 4.3.3的分析，应以后半“悲伤”的情感为主。然而我们选用的 CNN 模型中，CNN 的序列输出以最大池整合为一个固定长度的向量，即模型没有考虑情感出现的先后顺序，而是直接保留了字面上情感最强烈的内容，这也显示对语言模式的了解有助于模型的设

表 4.9　测试集上中间结果 II 对应的混淆矩阵

		预测标签			
		其他	开心	悲伤	愤怒
真实标签	其他	4467	65	60	85
	开心	82	197	4	1
	悲伤	38	1	204	7
	愤怒	47	1	5	245

表 4.10　测试集上最终结果对应的混淆矩阵

		预测标签			
		其他	开心	悲伤	愤怒
真实标签	其他	4478	60	57	82
	开心	82	197	4	1
	悲伤	39	1	203	7
	愤怒	47	1	5	245

计。

- **理解比喻修辞**。参考例子 3，用户甲在第一轮和第三轮中都在以比喻的修辞辱骂用户乙，然而系统从字面意思上未能识别出其情感倾向于“愤怒”。首先，在训练集上确实存在“be a pig”且大部分的标注都为“愤怒”，理论上机器学习算法可以学习出“be a pig”带有表达“愤怒”的性质。但数据中没有包含“be a dog”的样本，那么如果算法未能关联“pig”和“dog”就不可能识别出“be a pig”的情感属性。而从语言的角度来看，解决这个问题的本质是识别出比喻的使用，并且理解其真实想表达的意思，这就回归到上一章节中对修辞手法进行建模的问题。
- **语料中情感标注的不一致**。参考例子 4，第三轮中用户甲的发言仅包含两个大笑的表情符“😂”，而前两轮的文本内容则偏向中性，我们有理由认为这个样本应该标记为“开心”，然而组织者给出的标注为“其他”。再参考例子 5，第三轮中用户甲的发言同样仅包含大笑的表情符“😂”，而前两轮的文本内容同样偏向中性，但标注为“开心”。假如我们对例子 4 的情感判断正确，那就说明数据集内部对情感标注的标准存在不一致，这一方面会对模型的学习造成混淆，另一方面也会对系统评估带来影响。

4.6　本章小结

在本章中我们提出了一种多通道分类模型，并用于研究面向三轮对话的情感识别。我们针对国际比赛 SemEval-2019 任务三进行了多组实验，同时基于我们提出的系统设计方案，给出了一个具体的基于多步决策的情感识别系统。参赛结果显示我们系统的性能在 165 个参赛系统中排名前十。另外，基于对系统中间结果的分析，我们验证了我们系统中每一步决策的有效性，并解释它如何对系统的整体性能带来影响。最后我们对被系统错误识别的样本进行分析，指出系统的不足之处并给出对应的改进方向。

表 4.11　测试集中被系统误判的样本例子

编号	类别	对话
1	开心	（第一轮）用户甲: I’m in mood （第二轮）用户乙: ya need a hug ? :-) （第三轮）用户甲: yeah
2	悲伤	（第一轮）用户甲: I am （第二轮）用户乙: in your dreams （第三轮）用户甲: Haha Bite me! 😒😒😒
3	愤怒	（第一轮）用户甲: You are pig （第二轮）用户乙: so are you （第三轮）用户甲: You are dog
4	其他	（第一轮）用户甲: Wat u wanna knw? （第二轮）用户乙: i knw nothin :P （第三轮）用户甲: 😂😂
5	开心	（第一轮）用户甲: You cannot see my hair （第二轮）用户乙: I’m in your closet （第三轮）用户甲: 😂

第 5 章　总结和展望

5.1　论文工作总结

随着互联网用户越来越习惯于在社交媒体上发言，对这些数据进行内容分析的价值逐渐受到重视，相关研究得以蓬勃发展，面向文本的情感识别研究就是其中之一。本硕士论文探索了该领域下的两个核心问题：数据不均匀在多分类问题中的影响、在算法建模中引入上下文信息。为了解决以上两个问题，我们分别提出了一种基于多步决策的多分类系统框架以及一种多通道分类模型。本论文的主要研究工作如下：

1. **基于多步决策的微博反讽识别**。我们首先对和情感识别紧密相关的反讽识别进行研究，并指出在真实场景中数据不均匀的问题，为此我們提出了一种基于多步决策的多分类系統框架，把原本的多分类問題拆解成多个子分类問題的疊加，再透过逐步回答每个子分类問題得出最终的反讽识别结果。我们基于国际比赛 SemEval-2018 任务三的两个子任务进行实验。对于其中的子任务一，即识别微博文本是否带有反讽的二分类问题，我們訓練了一組以 2 层 BiLSTM 作为模型的二分类器，再透过多投票得出识别结果。实验结果显示我们的系统达到了当时参赛系统中靠前的性能，并和我们当时参赛的系统对比有了明显的进步。对子任务二，即在子任务一的基礎上把反讽细分成三个类别后的四分类问题，基于我们提出的系统框架，我们给出了一个具体的基于多步决策的反讽识别系统。实验结果显示我们系统在主要评价指标 F1 值上显着优于当时参赛系统中的第一，而在其他几项指标上都达到了仅次于第一名的数值，验证了我们系统的性能。另外透过对系统中间结果进行观察，我们解釋了多步决策中每一步如何对系統的整体性能帶來貢献，同时验证了我們系統框架的合理性。
2. **基于多通道模型引入上下文的情感识别**。为了研究如何在情感识别中引入上下文，我們研究了面向三轮对话的情感识别，同时提出了一种多通道分类模型，以结合起不同作用的上下文。我們基于国际比赛 SemEval-2019 的任务三进行实验。首先透过对语料进行观察，我们分析了三轮对话中每轮发言对识别最后一轮情感的作用，以此提出了一个三通道模型，再结合前面提出的系統框架，我們给出了一个具体的基于多步决策的情感识别系统。参赛结果显示我们的系统在 165 个参赛系统中排名前十。另外透过对系统的中间结果进行分析，我們验证了系統中每步决策的有效性，同时证明了基于多步决策的

系統框架通用于不同的多分类問題，但需要針对核心评价指标和各个类别的样本分布挑选合适的子分类問題。

5.2　未来工作展望

面向文本的情感识别技术在实际应用中有明显的不足之处，相关研究依然存在很大的进步空间。特别是随着信息技术发展，文本的使用场景只会变得越来越复杂。总结第3章和第4章我们对系统的错误分析，我们为面向文本情感识别的后续工作给出以下建议：

1. **多模型融合**。在两个章节的实验中，我们都指出了基于现阶段主流的人工神经网络可能无法同时对多个类别的数据进行拟合，原因在于各个类别背后的语言特征有本质上的不同。尝试提出拟合能力更强的人工神经网络结构固然是前进方向之一，但我们同时鼓励采用较简单的模型对局部数据进行建模，再透过类似我们提出的多步决策框架来结合各个模型捕足到的信息，以此解决单个模型建模能力有限的问题。。
2. **引入语义知识**。文本情感识别的核心难点之一在于语义。机器学习方法只能从训练数据中尝试学习词（组）的语义，但应用场景中往往会出现训练数据中没有出现过的词（组），导致算法无法理解其中的信息。为了处理这部分内容，我们建议引入更多额外的语义知识，以建立在训练数据中出现过和没有出现过的词（组）之间的联系。词嵌入方法虽然在原理上能关联语义相近的词，但从实验结果来看依然有不足之处（如无法处理上一词多义的情况），这些问题都有待进一步探索。
3. **引入语用知识**。文本情感识别的核心难点其二在于丰富多样的语用，譬如反讽和比喻等修辞手法。如果没有意识到发言者使用了特殊的修辞手法，单凭字面意思了解文本内容就有可能忽略甚至曲解发言者本身想表达的意思。而为了识别修辞的使用，本文中尝试采用多种人工神经网络模型进行建模，但在对系统进行错误分析时，我们会发现一些显而易见的模式未能被系统正确识别。对于这部分样本，最直观的做法是采用基于规则的方法。修辞手法的具体使用虽然多种多样，但根据语用知识我们可以总结出其中有限种模式。进一步地，有别于单纯基于规则的方法，我们建议结合基于规则的方法和机器学习方法，优先召回能被规则匹配的样本，再由机器学习方法识别难以由人工归纳出规则的样本，以此减轻人工设计规则的压力。

插图索引

表格索引

公式索引

参考文献

[1] Banerjee S, Dutta U. Detection of emotions in text: A survey: volume 03[M]. [S.l.: s.n.], 2015

[2] Chatterjee A, Narahari K N, Joshi M, et al. Semeval-2019 task 3: Emocontext: Contextual emotion detection in text[C]//Proceedings of The 13th International Workshop on Semantic Evaluation (SemEval-2019). Minneapolis, Minnesota: [s.n.], 2019.

[3] Alm C O, Dan R, Sproat R. Emotions from text: Machine learning for text-based emotion prediction[C]//Conference on Hlt/emnlp. [S.l.: s.n.], 2005.

[4] Plutchik R, Kellerman H. Emotion: Theory, research and experience[J]. In Theories of emotion, 1980, 11: 399.

[5] Ekman P. An argument for basic emotions[J]. Cognition & Emotion, 1992, 6(3-4): 169-200.

[6] Russell J. A circumplex model of affect[J]. Journal of Personality and Social Psychology, 1980, 39: 1161-1178.

[7] M. Bradley M, Greenwald M, C. Petry M, et al. Remembering pictures: Pleasure and arousal in memory[J]. Journal of experimental psychology. Learning, memory, and cognition, 1992, 18: 379-90.

[8] Watson D, Tellegen A. Toward a consensual structure of mood[J]. Psychological bulletin, 1985, 98: 219-35.

[9] Rubin D C, Talarico J M. A comparison of dimensional models of emotion: evidence from emotions, prototypical events, autobiographical memories, and words[J]. Memory, 2009, 17(8): 802-808.

[10] Mehrabian A. Pleasure-arousal-dominance: A general framework for describing and measuring individual differences in temperament[J]. Current Psychology, 1996, 14(4): 261-292.

[11] Hugo L. A new three-dimensional model for emotions and monoamine neurotransmitters[J]. Medical Hypotheses, 2012, 78(2): 341-348.

[12] Mohammad S, Bravo-Marquez F, Salameh M, et al. Semeval-2018 task 1: Affect in tweets[C]// Proceedings of The 12th International Workshop on Semantic Evaluation. [S.l.: s.n.], 2018: 1-17.

[13] Turney P D. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews[C]//Proceedings of the 40th annual meeting on association for computational linguistics. [S.l.]: Association for Computational Linguistics, 2002: 417-424.

[14] Pang B, Lee L. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts[C]//Proceedings of the 42nd annual meeting on Association for Computational Linguistics. [S.l.]: Association for Computational Linguistics, 2004: 271.

[15] Tang D, Qin B, Liu T. Learning semantic representations of users and products for document level sentiment classification[C]//Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers): volume 1. [S.l.: s.n.], 2015: 1014-1023.

[16] Khan A, Baharudin B, Khan K. Sentiment classification using sentence-level lexical based[J]. Trends in Applied Sciences Research, 2011, 6(10): 1141-1157.

[17] Li Y, Lin Y, Zhang J, et al. Constructing domain-dependent sentiment lexicons automatically for sentiment analysis[J]. Information Technology Journal, 2013, 12(5): 990-996.

[18] Dos Santos C, Gatti M. Deep convolutional neural networks for sentiment analysis of short texts[C]//Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers. [S.l.: s.n.], 2014: 69-78.

[19] Che W, Zhao Y, Guo H, et al. Sentence compression for aspect-based sentiment analysis[J]. IEEE/ACM transactions on audio, speech, and language processing, 2015, 23(12): 2111-2124.

[20] Wang Y, Huang M, Zhao L, et al. Attention-based lstm for aspect-level sentiment classification [C]//Proceedings of the 2016 conference on empirical methods in natural language processing. [S.l.: s.n.], 2016: 606-615.

[21] Pontiki M, Galanis D, Pavlopoulos J, et al. Semeval-2014 task 4: Aspect based sentiment analysis [C]//Proceedings of The 8th International Workshop on Semantic Evaluation (SemEval-2014). [S.l.: s.n.], 2014: 27–35.

[22] Tang D, Qin B, Liu T. Aspect level sentiment classification with deep memory network[J]. arXiv preprint arXiv:1605.08900, 2016.

[23] Tsur O, Davidov D, Rappoport A. Icwsm—a great catchy name: Semi-supervised recognition of sarcastic sentences in online product reviews[C]//Fourth International AAAI Conference on Weblogs and Social Media. [S.l.: s.n.], 2010.

[24] Davidov D, Tsur O, Rappoport A. Semi-supervised recognition of sarcastic sentences in twitter and amazon[C]//Proceedings of the fourteenth conference on computational natural language learning. [S.l.]: Association for Computational Linguistics, 2010: 107-116.

[25] Reyes A, Rosso P, Veale T. A multidimensional approach for detecting irony in twitter[J]. Language resources and evaluation, 2013, 47(1): 239-268.

[26] Kunneman F, Liebrecht C, Van Mulken M, et al. Signaling sarcasm: From hyperbole to hashtag [J]. Information Processing & Management, 2015, 51(4): 500-509.

[27] Littlestone N. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm[J]. Machine learning, 1988, 2(4): 285-318.

[28] Poria S, Cambria E, Hazarika D, et al. A deeper look into sarcastic tweets using deep convolutional neural networks[J]. arXiv preprint arXiv:1610.08815, 2016.

[29] Madhusudhanan S, Moorthi D M. A survey on sentiment analysis[J]. Indian Journal of Computer Science and Engineering, 2018, 9(2).

[30] 谢丽星, 周明, 孙茂松. 基于层次结构的多策略中文微博情感分析和特征抽取[J]. 中文信息学报, 2012, 26(1): 73-84.

[31] Go A, Huang L, Bhayani R. Twitter sentiment analysis[J]. Entropy, 2009, 17: 252.

[32] Paltoglou G, Thelwall M. Twitter, myspace, digg: Unsupervised sentiment analysis in social media[J]. ACM Transactions on Intelligent Systems and Technology (TIST), 2012, 3(4): 66.

[33] Khan F H, Bashir S, Qamar U. Tom: Twitter opinion mining framework using hybrid classification scheme[J]. Decision Support Systems, 2014, 57: 245-257.

[34] Angiani G, Ferrari L, Fontanini T, et al. A comparison between preprocessing techniques for sentiment analysis in twitter.[C]//KDWeb. [S.l.: s.n.], 2016.

[35] 张海涛, 王丹, 徐海玲, 等. 基于卷积神经网络的微博舆情情感分类研究[J]. 情报学报, 2018, 37: 695-702.

[36] Van Hee C, Lefever E, Hoste V. Semeval-2018 task 3: Irony detection in english tweets[C]// Proceedings of The 12th International Workshop on Semantic Evaluation (SemEval-2018). [S.l.: s.n.], 2018: 39-50.

[37] 刘丹丹, 邱恒清, 赵应丁. 基于 svm 的中文微博情感识别与分类研究[J]. 中国新通信, 2015, 17(21): 48-51.

[38] 邓钊, 贾修一, 陈家骏. 面向微博的中文反语识别研究[J]. 计算机工程与科学, 2015, 37(12): 2312-2317.

[39] Zahiri S M, Choi J D. Emotion detection on tv show transcripts with sequence-based convolutional neural networks[Z]. [S.l.: s.n.], 2017.

[40] Jackson P, Moulinier I. Natural language processing for online applications[M]. [S.l.]: John Benjamins, 2007

[41] Mitkov R. The oxford handbook of computational linguistics[M]. [S.l.]: Oxford University Press, 2004

[42] Ahmed F, Luca E W D, Nürnberger A. Revised n-gram based automatic spelling correction tool to improve retrieval effectiveness[J]. Polibits, 2009(40): 39-48.

[43] Nejja M, Yousfi A. The context in automatic spell correction[J]. Procedia Computer Science, 2015, 73: 109-114.

[44] Baziotis C, Pelekis N, Doulkeridis C. Datastories at semeval-2017 task 4: Deep lstm with attention for message-level and topic-based sentiment analysis[C]//Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017). Vancouver, Canada: Association for Computational Linguistics, 2017: 747-754.

[45] Kouloumpis E, Wilson T, Moore J. Twitter sentiment analysis: The good the bad and the omg! [C]//Fifth International AAAI conference on weblogs and social media. [S.l.: s.n.], 2011.

[46] Joshi A, Sharma V, Bhattacharyya P. Harnessing context incongruity for sarcasm detection[C]// Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers): volume 2. [S.l.: s.n.], 2015: 757-762.

[47] Baziotis C, Athanasiou N, Papalampidi P, et al. Ntua-slp at semeval-2018 task 3: Tracking ironic tweets using ensembles of word and character level attentive rnns[J]. arXiv preprint arXiv:1804.06659, 2018.

[48] Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space [J]. arXiv preprint arXiv:1301.3781, 2013.

[49] Deerwester S, Dumais S T, Furnas G W, et al. Indexing by latent semantic analysis[J]. Journal of the American society for information science, 1990, 41(6): 391-407.

[50] Pennington J, Socher R, Manning C D. Glove: Global vectors for word representation[C/OL]// Empirical Methods in Natural Language Processing (EMNLP). 2014: 1532-1543. http://www.aclweb.org/anthology/D14-1162.

[51] Cortes C, Vapnik V. Support-vector networks[J]. Machine learning, 1995, 20(3): 273-297.

[52] Drucker H, Burges C J, Kaufman L, et al. Support vector regression machines[C]//Advances in neural information processing systems. [S.l.: s.n.], 1997: 155-161.

[53] Rosenblatt F. The perceptron: a probabilistic model for information storage and organization in the brain.[J]. Psychological review, 1958, 65(6): 386.

[54] Rumelhart D E, Hinton G E, Williams R J. Learning internal representations by error propagation [R]. [S.l.]: California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[55] Hinton G E, Osindero S, Teh Y W. A fast learning algorithm for deep belief nets[J]. Neural computation, 2006, 18(7): 1527-1554.

[56] Basheer I A, Hajmeer M. Artificial neural networks: fundamentals, computing, design, and application[J]. Journal of microbiological methods, 2000, 43(1): 3-31.

[57] LeCun Y, Boser B, Denker J S, et al. Backpropagation applied to handwritten zip code recognition [J]. Neural computation, 1989, 1(4): 541-551.

[58] Waibel A, Hanazawa T, Hinton G, et al. Phoneme recognition using time-delay neural networks [J]. Backpropagation: Theory, Architectures and Applications, 1995: 35-61.

[59] Hubel D H, Wiesel T N. Receptive fields of single neurones in the cat's striate cortex[J]. The Journal of physiology, 1959, 148(3): 574-591.

[60] Hubel D H, Wiesel T N. Receptive fields and functional architecture of monkey striate cortex[J]. The Journal of physiology, 1968, 195(1): 215-243.

[61] Fukushima K. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position[J]. Biological cybernetics, 1980, 36(4): 193-202.

[62] Zhang X, Zhao J, LeCun Y. Character-level convolutional networks for text classification[C]// Advances in neural information processing systems. [S.l.: s.n.], 2015: 649-657.

[63] Elman J L. Finding structure in time[J]. Cognitive science, 1990, 14(2): 179-211.

[64] Hochreiter S, Schmidhuber J. Long short-term memory[J]. Neural computation, 1997, 9(8): 1735-1780.

[65] Graves A, Jaitly N, Mohamed A r. Hybrid speech recognition with deep bidirectional lstm[C]// 2013 IEEE workshop on automatic speech recognition and understanding. [S.l.]: IEEE, 2013: 273-278.

[66] Cho K, Van Merriënboer B, Gulcehre C, et al. Learning phrase representations using rnn encoder-decoder for statistical machine translation[J]. arXiv preprint arXiv:1406.1078, 2014.

[67] Schuster M, Paliwal K K. Bidirectional recurrent neural networks[J]. IEEE Transactions on Signal Processing, 1997, 45(11): 2673-2681.

[68] Mnih V, Heess N, Graves A, et al. Recurrent models of visual attention[C]//Advances in neural information processing systems. [S.l.: s.n.], 2014: 2204-2212.

[69] Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate[J]. arXiv preprint arXiv:1409.0473, 2014.

[70] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C]//Advances in Neural Information Processing Systems. [S.l.: s.n.], 2017: 5998-6008.

[71] Luong M T, Pham H, Manning C D. Effective approaches to attention-based neural machine translation[J]. arXiv preprint arXiv:1508.04025, 2015.

[72] Graves A, Wayne G, Danihelka I. Neural turing machines[J]. arXiv preprint arXiv:1410.5401, 2014.

[73] Xu K, Ba J, Kiros R, et al. Show, attend and tell: Neural image caption generation with visual attention[C]//International conference on machine learning. [S.l.: s.n.], 2015: 2048-2057.

[74] Joshi A, Bhattacharyya P, Carman M J. Automatic sarcasm detection: A survey[J]. ACM Computing Surveys (CSUR), 2017, 50(5): 73.

[75] Hazarika D, Poria S, Zadeh A, et al. Conversational memory network for emotion recognition in dyadic dialogue videos[C]//Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). [S.l.: s.n.], 2018: 2122-2132.

致　谢

衷心感谢导师徐明星副教授多年来对本人的指导和帮助，使我在研究生期间的三年受益匪浅。徐老师在科研问题上有独特的想法，对于我主要研究的自然语言处理领域，当很多研究都专注于采用更复杂的模型进行尝试时，徐老师强调从语言的本质入手，除了亲自翻查语言学的相关材料，还会向心理学的老师讨论交流，这种对研究的热情和好奇心对我有很大的感染力。除了学术以外，徐老师还会有一些为人处事方面的教导，确实做到了授业解惑。在此再次表示由衷的感谢。

另外感谢在读期间帮过我的各位老师和同学，因为有你们的帮助使得我得以顺利完成我的硕士学业。感谢家人在背后的支持，让我远离家乡也没有顾虑。最后感谢每一位和我非常亲密的好友，是你们的支持帮我度过了这些艰难的时刻。

最后再次向以上的各位表示衷心的感谢。

声　明

本人郑重声明：所呈交的学位论文，是本人在导师指导下，独立进行研究工作所取得的成果。尽我所知，除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人享有著作权的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。

签　名：_____ 日　期：_____

个人简历、在学期间发表的学术论文与研究成果

个人简历

1994 年 6 月 4 日出生于澳门特别行政区。

2012 年 9 月考入清华大学计算机科学与技术系，2016 年 7 月本科毕业并获得工学学士学位。

2016 年 9 月免试进入清华大学计算机科学与技术系攻读硕士学位至今。

发表的学术论文

[1] Xihao L, Ye M, Mingxing X. THU-HCSI at SemEval-2019 Task 3: Hierarchical Ensemble Classification of Contextual Emotion in Conversation [C]//Proceedings of The 13th International Workshop on Semantic Evaluation (SemEval-2019). [S.l.: s.n.], 2019: 345-349.

综合论文训练记录表

学生姓名		学号		班级	
论文题目					
主要内容以及进度安排	指导教师签字： 考核组组长签字： 年　　月　　日				
中期考核意见	考核组组长签字： 年　　月　　日				

指导教师评语	指导教师签字：_____ 年　　月　　日
评阅教师评语	评阅教师签字：_____ 年　　月　　日
答辩小组评语	答辩小组组长签字：_____ 年　　月　　日

总成绩：_____

教学负责人签字：_____

年　　月　　日