社会科学进展
2024 年 12 月第 6 卷第 6 期

基于深度学习模型的正负向文本情感分类

邓依霖　徐舒文　高　昕

江苏师范大学，徐州

摘　要 | 本文结合文本情感分类的核心目标和研究现状，提出了一种基于深度学习的文本情感分类方法。通过对传统机器学习方法的总结与评估，分析其在处理复杂情感表达时的局限性；运用深度学习方法，搭建了一个正负情感分类模型，可用于微博评论的数据分析与测试。该深度学习模型基于正负情感的文本分类任务，对海量文本数据实现预处理和训练测试。实验结果表明，批处理次数在1000次以内，测试集可达平均98%的准确度；对十万多条正负情感文本，可实现准确分类。与此同时，作者针对文本情感分类难题，如结构不良与讽刺文本、粗粒度情感分析、文化意识缺乏、依赖数据和注释，以及词嵌入局限性等，作了总结性评述和展望。

关键词 | 文本情感分类；自然语言处理；深度学习；情感倾向；微博

1　引言

网络信息化时代正以前所未有的蓬勃发展态势，影响着人类生活与科技进步。与此同时，社会媒体也呈现出多样化形态，论坛、博客、微信和微博等媒介日益发达。随着用户参与度的提高，人类对于网络的使用方式，也产生了巨大变革。用户不再是被动获取网络知识，而是更积极地创作各类信息。因此，在网络媒介中，呈现出各类形式的大量主观性信息，用来表达用户情感、情绪和观点，以文本信息为主体表现形式。自然语言处理（NLP）领域中，针对这些信息，如何更有效地利用数据，提取人们感兴趣的、携带观点的文本，并且对其做出准确分析，代表一项非常重要的研究课题[1]。

伴随互联网的普及，社交媒体应用日新月异，海量文本数据，在全球范围内迅速集聚。如何从这些数据中，提取有价值的信息，成了现代大数据分析和自然语言处理领域的重要课题。在这些任务中，文本情感分类（Sentiment Analysis），通过自动化手段，对文本情感倾向进行识别和分类，这类技术愈发受到广泛关注。应用文本情感分类工具，可以有效地挖掘和分析人们观点、情绪、评论和态度，进而推断针对诸如产品、服务、组织、个体、事件、主题等实体的情感倾向，再进一步归纳、推理和提炼其中的有用信息。它代表着一个庞大的问题空间，也意味着拥有众多研究内容。目前，该研究领域，主要包括文本情感信息分类、文本情感信息抽取，以及文本情感分析技术的应用[2-3]。在文本情感信息分类研究中，非常典型而重要的几项课题，包括主客观分类、褒贬情感倾向判别和强度分类[3]；在信息抽取方向，主要研究任务，包括观点句中的几类相关要素（例如观点持有者、评价词或情感词，以及评价搭配

基金项目：江苏师范大学博士学位教师科研启动项目（22XFRS034）。
通讯作者：高昕，江苏师范大学物理与电子工程学院讲师，研究方向：人工智能、智能图像处理、目标检测与分类、数学建模、统计信号处理等。
文章引用：邓依霖，徐舒文，高昕．基于深度学习模型的正负向文本情感分类［J］．社会科学进展，2024，6（6）：1397-1405.
https://doi.org/10.35534/pss.0606155

等）的抽取。此外。在情感分析技术应用领域，则主要是基于情感信息分类以及信息抽取为基础[4]。

尽管文本情感分类在多个领域取得了显著进展，但由于文本的情感倾向，通常具有多样性和复杂性，情感分类任务，仍然面临着许多挑战。首先，文本中的情感表达，往往依赖于上下文信息，且存在着不少难以直接从字面理解的情感表达方式，例如讽刺、双关、反语等，这就对传统的情感分类模型，提出了更高要求。其次，情感的判定，不仅依赖于单词层面的情感色彩，还需要考虑句子的结构、语法，以及上下文中的隐含信息；这使得在处理长文本时，对情感分类任务的实现，变得更为复杂。此外，由于情感词汇本身具有高度的多义性，同一个词汇在不同语境下，所需表达的情感含义，可能完全不同，这也增加了情感分析的难度。

近年来，综观文本情感分类课题的研究进展，深度学习技术得到了广泛应用。传统的情感分类方法，主要分为两类：一类依赖于手工提取特征，如情感词典和词频–逆文本频率指数（TF–IDF）等；另一类借助经典机器学习理论，如支持向量机、朴素贝叶斯算法等。这些方法在情感分类中，已取得了一些成果；然而，由于无法有效捕捉文本中的复杂语义和上下文信息，往往遇到一些难题，如泛化能力差，以及对复杂情感表达理解不足等。而基于卷积神经网络的深度学习模型，通过端到端的训练方式，能够自动学习文本中的高层次特征，克服了手工特征设计的局限性。

本文旨在探讨基于深度学习的文本情感分类方法，根据深度学习理论，搭建相应模型，用于情感分类。作者采用了微博中带有情感标注的数据，重点分析深度学习模型在情感分类中的准确度；介绍当前主流的深度学习方法，特别是基于神经网络的情感分类模型，探讨其在实际应用中的效果与挑战；展示实验所用的数据集，评估模型在公开情感分析任务上的表现，并分析不同方法的优势与适用场景；同时，论文展望了当前情感分析领域的开放性难题，如情感词汇的不确定性、讽刺和双关的处理，以及多语言情感分类等。先前代表性学者中，王钦扬[5]等详细介绍了不同研究中的分类技术、采纳的数据集和实验结果。燕道成[6]等通过跟踪日本排污治污的评论，开展文本情感分析，探究影响受灾群众情绪的因素，以及网民情绪倾向特征。杜明利[7]等借助LDA（Latent Dirichlet Allocation）主题模型，剖析评论文本，挖掘其中的隐性主题和话题分布，并进一步揭示用户对产品或服务的不同观点和需求，从而为京东商城的发展，有针对性地提供改进策略和建议。

人工智能时代，在智能客服、舆情监控、个性化推荐等服务性行业领域，情感分析的应用前景广阔。然而，为了实现更高效的情感分类，获取更精准的识别率，研究学界依然需要付出更艰巨的努力，主要体现在优化深度学习模型、完善多模态情感分析算法，以及改进跨语言情感分类方案等多方面内容。

2 理论介绍

2.1 文本情感分类研究现状

与国外研究相比，中国大陆对于微博文本的研究，起步较晚。基于微博文本情感分类的参考文献和调研报告，作者通过细心搜索，并仔细研读后发现，基于微博文本的情感分析，大体可分为三类：文本的预处理（如分词等操作）、特征抽取与特征选择，以及分类算法[8]。至于微博文本语料库，先前的研究，多数直接进行二分类（积极或消极），或是三元（多加一个客观无情感）分类。对应的分类方法，大致可分为两类：一类以情感词典为核心，另一类以特征选择和特征提取为核心，实现机器学习。现进一步研讨如下。

2.1.1 以情感词典为核心的情感极值计算

传统的基于情感词典的分类模型，在对微博文本语料库进行分类研究时，其核心重点在于如何创建和扩充用户所使用的情感词典。针对情感词典等课题，2008年，林鸿飞等[9]相关学者，在研究成果已颇具规模的基础上，构建了一个中文情感词语本体库，依据对人类情感的细致划分，将词语的情感类别，归结为七大类。对于情感词典的自动更新，2006年，朱嫣岚等[10]基于知网HowNet词汇，实现相似度计算。由于微博文本内容复杂多样，2013年，侯敏等[11]提出“将短语更新到情感词典当中”的工作理念。考虑到二分类模型的缺陷，2012年，孙建旺等[12]提出了综合情感词典和机器学习的方法，选取形容词和动词作为分类特征，使用情感词典来计算特征权值大小，最后用支持向量机（SVM）实现分类。此外，党蕾[13]采用否定模式匹配和依存句法分析，探讨了依存语法距离的方法；赵研研等[14]提出了一种基于短语句法分析的方法；Wiebe[15]在标记的种子词基础上，采用聚类算法，针对各类未经标注的形容词词汇，实现准确分类。

2.1.2 基于深度学习的情感分类

2006年，辛顿（Hinton）等[16]学者发表了关于深度信念网络的论文，提出采用深度结构训练神经网络的方法。随后，米科洛夫（Mikolov）曾在 2010 年将递归神经网络用于自然语言处理[17]，三年后又部署了Continue BOW（CBOW）模型和Skip–gram模型[18]。杰弗瑞（Jeffrey）和索赫尔（Socher）于2014年提出了Glove模型[19]，随即将深度学习模型用于匹配自然语言处理的工作任务，比如机器翻译[20–21]和信息检索[22–23]。

深度学习是机器学习的子集，也是目前情感分类的研究热点之一。由于使用机器学习相关的技术，需要在前期进行监督训练，即从大量高质量带标签的语料库中，抽取相关特征，再通过分类模型，训练一个分类

器。目前，主流的机器学习分类器，采用最大熵、K近邻（KNN）、SVM和朴素贝叶斯（NB）等浅层学习算法，面对各种微博文本语料库，都可以实现情感二分类。然而，处理高维、海量、复杂的多模态数据，则需要通过神经网络，自动学习特征和规则，利用循环神经网络（RNN）和长短期记忆网络（LSTM）等结构，妥善处理序列数据，代表深度学习在自然语言处理中的主要工作场景[24–25]。2015 年，辛顿（Hinton）还在《自然》（*Nature*）期刊上撰文表示，“未来几年内，在自然语言处理等领域上，深度学习会发挥重大作用”[26]。

2.2 核心算法原理

作者在设计文本情感分类模型的核心算法时，采用长短期记忆网络（LSTM），现对其主体结构概括如下[27]。

（1）LSTM：长短期记忆网络（long short-term memory network，LSTM）是一种特殊类型的循环神经网络（RNN），专门为深度学习而设计，用来解决在长序列数据学习中，传统 RNN所遇到的梯度消失，或梯度爆炸等问题。LSTM 网络的基本单元，包含了四个重要组件：遗忘门（Forget Gate）、输入门（Input Gate）、更新门（Cell Update）和输出门（Output Gate）。LSTM 的核心架构，是引入了记忆单元和门控机制，通过这些机制，使得神经网络能够捕捉到长期依赖关系，记住长时间跨度信息。LSTM 的核心思想，在于通过引入门控机制，对每一个时间步的记忆进行控制。每个门控机制的输出，都是一个0～1之间的数值，表示对相应信息的保留程度。为此，LSTM 能够在学习过程中，根据不同情境，决定是否保留信息。

（2）Embedding（词嵌入）：词嵌入技术将词语转换为向量表示，使得机器可以理解和处理自然语言文本。自然语言处理中的词嵌入方法，通过将每个词语映射到一个高维空间中的稠密向量，表示该词语的语义和语法特征。这种表示不仅能够捕捉词语之间的语义相似性，还能帮助提高任务的准确性，如文本分类、情感分析、机器翻译等。词嵌入的核心思想是将某一单词表示成可以反映该词上下文信息的一个向量。词语的相似性和关系可以通过向量空间的距离来衡量；而对于近义词而言，语义越相近，其向量应该更接近。词嵌入的生成主要可以通过计数和预测两类方法来实现。将词语映射到向量空间，促使计算机更好地理解自然语言中的语义和语法关系，体现词嵌入技术在NLP中的技术价值。

3 数据获取与预处理

3.1 数据集

微博作为中国最大的社交平台之一，用户日常发表的评论，涵盖了丰富的主题，蕴藏着多样化的情感表达。通过对这些评论进行情感分析，不仅能够揭示公众对特定事件、产品或话题的情感倾向，也能为企业、政府和社会研究机构，提供宝贵的舆情数据和决策依据。

作者搜集了来自微博平台的评论，采用文本情感分析法，进行情感倾向分析。这些带情感标注的评论，共计10万多条，其中正负向评论各约5万条，保存在后缀为.csv的文件中，随后进行相应处理。

3.2 数据预处理

首先，需要从.csv文件中，加载带有情感标注的中文社交媒体数据。数据文件包含两列：text（社交媒体文本内容）和label（情感标签）。其中，label分为0（负面情感）和1（正面情感）。对汉字进行文本处理的第一步是分词。由于中文文本由连续的汉字组成，不像英文那样有空格分隔单词，故首先应进行分词处理。

目前，结巴分词（Jieba）是一种非常流行的中文分词工具[28]，使用它对每条文本进行分词，可将文本转换为词语列表。所谓的停用词，是指那些在文本分析中无太大意义的词汇，如“的”“是”“了”这样的常见词。在文本处理中，这些无意义的词汇，通常会预先剔除掉，以减少噪声。

作者使用pandas库读取相应数据，并使用结巴分词处理中文文本，排除停用词（如常见的无意义词）。随后创建词汇字典，并记录每个词语的出现频率。实际应用中，常常会事先设置一个阈值，滤除出现频率较低的词，从而减少字典的大小。最后，生成词汇字典，在记录词频的同时，创建未知词（UNK）和填充词（PAD）的映射。数据预处理后，每条词汇的数据标签，均与索引内容逐一对应，结果如图1所示。

{'泪': 0, '~': 1, '嘻嘻': 2, '都': 3, '爱': 4, '抓狂': 5, '鼓掌': 6, '不': 7, '…': 8, '好': 9, '人': 14, '衰': 15, '还': 16, '晕': 17, '说': 18, '吃': 19, '太': 20, '偷笑': 21, '很': 22, '没': 27, '想': 28, '上': 29, 'http': 30, 'cn': 31, 't': 32, '北京': 33, '开心': 34, '可爱': 39, '没有': 40, '做': 41, '心': 42, '不是': 43, '中国': 44, '转发': 45, '最': 46, '现在': 1, '汗': 52, '真': 53, '赞': 54, '谢谢': 55, '才': 56, '月': 57, '真的': 58, '蜡烛': 59, '到': 64, '给力': 65, '已经': 66, '请': 67, '一起': 68, '走': 69, '酒店': 70, '更': 71, '旅': '老': 76, 'good': 77, '围观': 78, '一下': 79, '哈哈哈': 80, '老师': 81, '年': 82, '后': 83, '呵呵': 88, '不要': 89, '孩子': 90, '吃惊': 91, '一': 92, '亲亲': 93, '2': 94, '花心': 95, '时间': 100, '失望': 101, '感谢': 102, '悲伤': 103, '生活': 104, '酷': 105, '带': 106, '希': '幸福': 111, '新': 112, '找': 113, '1': 114, '上海': 115, '出来': 116, '终于': 117, '明天': '可怜': 122, '快乐': 123, '死': 124, '这是': 125, '期待': 126, '活动': 127, '发现': 128, '应该': 133, '3': 134, '只': 135, '完': 136, '最后': 137, '必须': 138, '看看': 139, '儿': '天': 144, '抱抱': 145, '家': 146, '控': 147, '伤心': 148, '天': 149, '一次': 150, '怒骂': '好吃': 155, '旅行': 156, '事': 157, '话': 158, '鼻屎': 159, '回来': 160, '5': 161, '亲': '家': 166, '很多': 167, '蛋糕': 168, '钱': 169, '那': 170, '滴': 171, '继续': 172, '手机': '方': 177, '准备': 178, '餐厅': 179, '听': 180, '送': 181, '问题': 182, '号': 183, '出': 18 '求': 188, '前': 189, '工作': 190, '加油': 191, '恭喜': 192, '美': 193, '咖啡': 194, '猫': '玩': 199, '10': 200, '分享': 201, '东西': 202, '最近': 203, '推荐': 204, '需要': 205, '原 209, '我爱': 210, '每天': 211, '周末': 212, '买': 213, '写': 214, '特别': 215, '网': 216, 0, '官方': 221, '问': 222, '店': 223, '话筒': 224, '7': 225, '两个': 226, '张': 227, '8': '屈': 232, '睡觉': 233, '已': 234, '却': 235, '王': 236, '僵': 237, '穿': 238, '只能': 239, '公司': 244, '味道': 245, '看来': 246, '礼物': 247, '人生': 248, '美女': 249, '男人': 250, 254, '下次': 255, '见': 256, '童鞋': 257, '新浪': 258, '生病': 259, '摄影': 260, '刚': 261

图 1 部分词汇的数据标签和对应索引

Figure 1 Data labels and corresponding indices of some vocabularies

4 实验研究内容

4.1 模型搭建

传统意义上，基于情感词典实现文本情感分类的流程，如图2所示，通常包括预处理、自动分词、训练情感

https://doi.org/10.35534/pss.0606155

词典，并基于一定的判断规则，实现自动分类等步骤。①

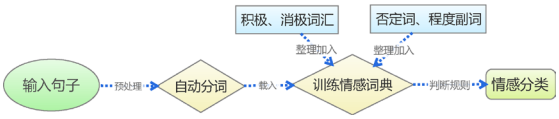

图 2　基于情感词典实现文本情感分类的流程图

Figure 2　Block diagram on classification of text sentiment based on sentiment dictionary

对长文本文档做情感倾向性分析时，通常采用基于注意力机制的双层LSTM，基本原理如图3所示[29]：首先，对句子级的情感向量表示，利用LSTM来表征学习。随后考察各种句式的情感语义表达，以及句子间的语义关系，并通过双向LSTM实现编码；根据句子之间情感语义贡献度的差异，基于注意力机制，合理分配其权值。最后，对所表示的情感向量，经加权处理后，实现文档级的情感向量表示，再经过Softmax层转换后，输出对应的长文本情感倾向。

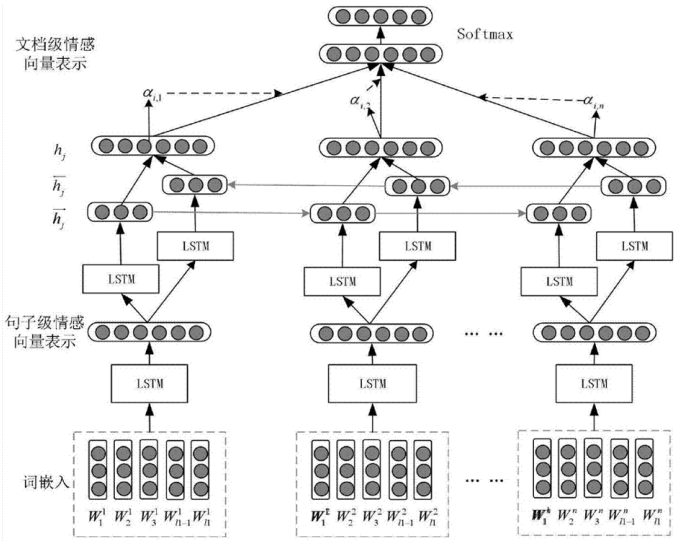

图 3　基于注意力机制的双层LSTM长文本情感倾向性分析[29]

Figure 3　Analysis of sentiment tendencies of long texts via attention mechanism based Bi-LSTM

本项实验中，我们构建了一个深度学习模型，旨在完成文本情感分类任务。模型架构主要依赖于词嵌入（Embedding）、双向长短记忆网络（LSTM）、卷积层和全连接层，进而处理输入的文本数据，并进行准确分类。作者在基于注意力机制的双层LSTM基础上，对网络结构作了相应修改，核心代码如图4所示。

```
class Model(nn.Module):
    def __init__(self, config):
        super(Model, self).__init__()
        # 词嵌入层
        self.embeding = nn.Embedding(config.n_vocab,
                                     embedding_dim=config.embed_size,
                                     padding_idx=config.n_vocab - 1)
        # lstm层
        self.lstm = nn.LSTM(input_size=config.embed_size,
                            hidden_size=config.hidden_size,
                            num_layers=config.num_layers,  # lstm的层数
                            bidirectional=True,  # 双向lstm层
                            batch_first=True,
                            dropout=config.dropout)  # 防止过拟合
        # 卷积层
        self.maxpooling = nn.MaxPool1d(config.pad_size)
        # 全连接层
        self.fc = nn.Linear(config.hidden_size * 2 + config.embed_size,  # 双向LSTM
                            config.num_classes)

        self.softmax = nn.Softmax(dim=1)
```

图 4　模型部分代码展示的输出图

Figure 4　Output snapshot of displayed code segments on the model

① https://cloud.tencent.com/developer/article/1061217。

以上深度学习模型，适用于文本情感分类。现对模型中实现的主要网络结构进行简要说明[①]。

（1）词嵌入层

词嵌入层作为模型的起始部分，将词汇表中的词映射为固定大小的稠密向量。在init方法中，self.embedding是一个Embedding层，它以词汇表大小为输入，以每个词的向量表示为输出，用config.embed_size来指定相应的维度。此外，padding_idx的设置值为config.n_vocab-1，意味着在输入序列中填充的词（通常是零），不会影响模型的学习过程。

（2）LSTM层

双向LSTM层以self.lstm为表征，其输入为词嵌入后的向量，输出则是经过LSTM处理后的隐藏状态。该层从前后两个方向，同时对文本序列进行处理，捕获了更多上下文信息。此外，LSTM的各个参数（如input_size、hidden_size、num_layers、dropout等），都来自config配置。注意到程序代码中，LSTM输出out的尺寸为［batch_size，seq_len，hidden_size * 2］，因为双向输出的维度数目，会是隐藏层大小的两倍。

（3）拼接与激活层

LSTM完成输出之后，out＝torch.cat［（embed，out），2］，将原始的词嵌入向量（embed）和LSTM的输出（out），在最后一个维度上进行拼接。拼接后的out，对应的张量尺寸为［batch_size，seq_len，hidden_size*2+embed_size］。这意味着每个词的表示，同时包含了原始嵌入和LSTM对它的上下文理解。接着，调用F.relu（out），将拼接后的结果，通过ReLU激活函数来表征，使得模型能够引入非线性，增加网络泛化能力。

（4）卷积和池化层

随后，out = out.permute（0，2，1），交换张量的维度，将out从［batch_size，seq_len，hidden_size*2+embed_size］转换为［batch_size，hidden_size*2＋embed_size，seq_len］。接着，self.maxpooling（out）通过一维卷积（MaxPool1d），可以对特征进行池化操作。这里使用的是最大池化操作，作用是提取序列中最显著的特征。该操作将序列长度（seq_len）压缩为一个较小的尺寸，从而减少了模型的计算负荷，并加强特征的表达能力。池化后输出的结果，被重新调整为二维张量out.size（）［0］，－1，并且通过全连接层进行分类。

（5）全连接层和Softmax层

最后，拼接后的中文文本特征，通过全连接层（self.fc）进行映射。此时，输出的维度即为类别数（config.num_classes），代表执行文本分类任务的最终结果。

为了将输出转化为概率分布，模型通过self.softmax（out），应用Softmax函数，得到每个类别的概率值。

4.2 模型训练

在模型训练阶段，我们搭建了上述模型，并装载了预处理好的数据，在Python测试平台上进行训练。首先，通过读取指定路径的数据集文件和词汇表文件，构建自定义的文本分类数据集，并生成数据加载器，用于批量加载训练数据。随后，作者对代码配置了一些超参数，对主要的模型参数，现说明如下：

该双层LSTM，词汇表大小为50000，以input_dim表示；嵌入向量的维度为64，以embedding_dim表示；LSTM层的神经元数量为128，通过lstm_units来显示；输出层的尺寸，通常与词汇表大小等同，即input_dim = 50000。

设置学习率为0.01，训练轮数设置为100，并初始化模型，迭代输出的部分结果，如图5所示。

```
        optimizer.step()  # 更新参数

    if epoch % 10 == 0:
        torch.save(model_text_cls.state_dict(), "../models/{}.pth".format(epoch))
```

```
epoch is 0, ite is 919, val is 0.31328827142715454
epoch is 0, ite is 920, val is 0.32886916399002075
epoch is 0, ite is 921, val is 0.3132988512516022
epoch is 0, ite is 922, val is 0.339650422334671
epoch is 0, ite is 923, val is 0.35234662890434265
epoch is 0, ite is 924, val is 0.31328484416007996
epoch is 0, ite is 925, val is 0.3289182782173157
epoch is 0, ite is 926, val is 0.32109785079956055
epoch is 0, ite is 927, val is 0.3445505201816559
epoch is 0, ite is 928, val is 0.33677977323532104
epoch is 0, ite is 929, val is 0.32109755277633667
epoch is 0, ite is 930, val is 0.3133189082145691
epoch is 0, ite is 931, val is 0.3445359766483307
epoch is 0, ite is 932, val is 0.32891666889190674
epoch is 0, ite is 933, val is 0.3289642632007599
epoch is 0, ite is 934, val is 0.3524179458618164
epoch is 0, ite is 935, val is 0.3291010856628418
epoch is 0, ite is 936, val is 0.3132941424846649
epoch is 0, ite is 937, val is 0.3326720595359802
```

图 5　部分迭代展示的输出图

Figure 5　Output snapshot of displayed iteration segments

① https://github.com/yingzhang123/Text_Sentiment_Classification/tree/master/。

https://doi.org/10.35534/pss.0606155

模型结构和参数由Config配置文件提供。在该阶段，模型通过输入的文本数据，进行前向传播，计算出预测结果，并调用交叉熵损失函数，衡量模型预测值与实际标签之间的误差，其表达式如下[27]：

$$loss = -\sum_{s\in S}\sum_{c=1}^{C} P_c^g(s)\cdot log(P_c(s)) \qquad (1)$$

其中，S和C分别表示训练数据与情感类别数，s代表某一句话；通过Softmax层，给出预测s为C类的概率，表征为$P_c(s)$；$P_c^g(s)$是一项输出为1或0的数值，表示C类的情感类正确与否。通过反向传播，计算损失函数对模型参数的梯度，并利用随机梯度下降等优化器，更新模型参数，优化模型性能。训练过程中，输出每个训练周期和批次的损失值，便于监控训练进展。

在训练完成一定轮次后，代码会定期保存模型参数，以便后续使用。整个工作流程，包括数据准备、模型训练、参数更新和模型保存四个阶段。

5 实验结果与分析

5.1 仿真测试结果

为了验证该文本情感分析模型的准确度，我们将模型运用于测试集上，设定批处理的Batch为1000次。模型测试的准确度，其量化输出结果，如图6所示。由此可知，该模型准确度较高，主要集中在0.97～0.99之间。

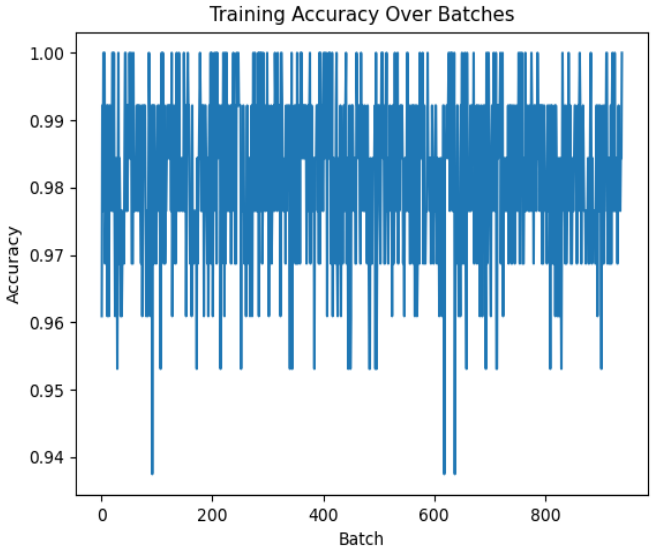

图 6 测试集准确度的量化输出

Figure 6 Quantitative outputs on the accuracy of test dataset

为了进一步验证模型的准确度，该数据集中的文本，可以直接输入模型来测试，让模型分析该文本的情感，判断其情感倾向是正向还是负向；对应的测试代码段和情感预测结果，分别如图7和图8所示。

```
voc_dict = read_dict(dict_path)
stop_words = open(stop_words_path, encoding='utf-8').readlines()
stop_words = [word.strip() for word in stop_words]  # 去除换行符

# 加载训练好的模型
model = Model(cfg)
model.load_state_dict(torch.load("C:/Users/Dengy/Desktop/Text_Sentiment_Classification-master(2)/Text_Sentiment_Classification-master/
model.to(cfg.devices)

# 测试文本
test_text = "梦想有多大，舞台就有多大![鼓掌]"  # 示例文本

# 使用模型进行情感预测
sentiment = predict_sentiment(test_text, model, voc_dict, stop_words, cfg.pad_size, cfg.devices)

# 输出预测结果
if sentiment == 0:
    print("情感预测：负面")
else:
    print("情感预测：正面")
```

```
预测的softmax概率： tensor([[2.5537e-14, 1.0000e+00]])
情感预测：正面
```

图 7 正向情感文本测试输出

Figure 7 Test output on text-sentiments of positive tendency

```
voc_dict = read_dict(dict_path)
stop_words = open(stop_words_path, encoding='utf-8').readlines()
stop_words = [word.strip() for word in stop_words]  # 去除换行符

# 加载训练好的模型
model = Model(cfg)
model.load_state_dict(torch.load("C:/Users/Dengy/Desktop/Text_Sentiment_Classification-master(2)/Text_Sentiment_Classification-master/
model.to(cfg.devices)

# 测试文本
test_text = "这是怎么了... //@chicphered114:大晚上能不发这么感人的吗？ //@尚龙申君: 命运... //@刘璐瑶同学:真的很感人~[泪][泪][泪]"  #

# 使用模型进行情感预测
sentiment = predict_sentiment(test_text, model, voc_dict, stop_words, cfg.pad_size, cfg.devices)

# 输出预测结果
if sentiment == 0:
    print("情感预测：负面")
else:
    print("情感预测：正面")
```

```
预测的softmax概率： tensor([[1.0000e+00, 1.6085e-17]])
情感预测：负面
```

图 8　负向情感文本测试输出

Figure 8　Test output on text-sentiments of negative tendency

由以上测试结果可知，该模型在所用的测试集内，文本情感测试符合预期结果，并且准确度较高，能正确区分出文本情感的正负向。

5.2　文本情感分析

情感分析可以通过情感极性判断与程度计算，确定文本情感倾向。常用方法包括基于情感词典的方法和基于机器学习的方法。基于情感词典的方法，依赖于预构建的情感词典，通过匹配文本词语与词典情感词，来判断情感极性；基于机器学习的方法，则利用标注好的训练数据，训练情感分类模型，对新文本进行情感预测。

根据微博中的评论，作者将文本情感分析的具体内容，简要归结为以下几方面。

（1）词汇识别：评论中出现了积极词汇，例如“非常棒”“清晰”“好”“快”“喜欢”等。

（2）情感极性判断：观察上述词汇，整体的情感倾向，应该是积极而乐观的。

（3）情感程度评估：从“非常棒”等描述可以看出，该评论对产品评价非常高，属于非常积极的情感程度。

通过上述文本情感分析，可以得出结论，该评论表达了正面情感倾向。对于各大网络平台，这有助于了解用户对产品或事件的满意度和反馈，从而为改进产品质量和制定营销策略，提供切实可靠的依据。

5.3　工作局限点与展望

在文本情感分析领域，尽管目前的课题研究，已经取得了一定进展，但还存在不少局限，现概括如下。

（1）结构不良与讽刺文本。这类文本具有拼写错误、语言不规范等缺陷，并且语法复杂。采用传统情感分析方法，难以给出准确解释。至于讽刺文本，往往以言外之意、反语或夸张形式表达，需要更深层次的语境理解。单纯应用传统情感分析模型，则难以捕捉其隐含情感，甚至可能会导致情感误判。

（2）粗粒度情感分析。情感分析具有很强的主观性。缺乏细粒度的考量，会限制人类对文本情感的深入理解，通常只能划分为几大类别，如积极、消极与中性，导致无法明确捕捉情感的强度变化。在同一类别下，文本可能包含的细微差异，可能会被忽略；用户的真实情感，识别准确度受限。此外，由于情感种类多元化，过于简单的情感分类，容易忽略文本语境对情感表达的影响，可能导致对情感含义的误判。

（3）文化意识匮乏。处理不同文化或地域的文本时，情感分析模型可能会因为尚未全面考虑特定文化背景，导致情感判断误差。由于情感表达方式受到文化差异制约，在不同文化语境下，同一表达，可能代表不同情感含义。设想在跨文化场景中，如果文本情感分类模型缺乏文化意识，可能就难以理解与解释。

（4）依赖注释数据。基于深度学习的情感分析模型，通常需要大量标注数据，实现特征训练与模块化评估，导致其对特定领域具有依赖性，或囿于标注者的主观判断，限制了它们在其他领域或不同群体中的泛化能力。为了获取大规模高质量注释数据，算力资源与执行时间的过度耗费，成为情感分析模型拓展到新语言和新领域的主要障碍。

（5）词嵌入受限。在捕捉文本中词语及其含义之间的复杂关系等方面，词嵌入具有一些局限性；这可能会导致模型泛化能力不足，难以始终如一地准确表示文本中所表达的情感。

6　结论

作为自然语言处理领域的关键分支，文本情感分析模型以及情感倾向分类测试，不仅对信息检索、社交媒体分析、消费者行为研究等多个领域，产生着深刻影响，而且也为人类情感动态的理解，提供了强有力的工具。

现阶段，越来越多的用户，在社交媒体上发表自己的观点看法，表达内在情感。通过分析其情感极性，就

https://doi.org/10.35534/pss.0606155

可以判断他（她）们的态度。特别是在对于政治立场、网络购物等问题的评论中，客观而理性的情绪分析，十分有必要。通过判断其情感极性，可以未卜先知，预测将来可能发生的事情。由于网络评论、微博、博客等社交媒体中的文本蕴含着大量的情绪元素，情感分析是对文本中的情感进行计算，对情绪进行数据挖掘、分析判断和测试验证。这在人工智能时代，心理健康评估与情绪管理领域不仅代表一项意义深远的挑战性工作，也会对智能化的情感理解和驱动未来产生积极而有益的影响。

致谢

作者衷心感谢江苏师范大学语言科学与艺术学院马勇副教授悉心讲授“自然语言处理”课程，同时对本项课题研究提供支持与帮助。

参考文献

[1] X J Wan. Co-Training for cross-lingual sentiment classification [C]. Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP. Suntec, Singapore, 2009: 235-243.
[2] 谢丽星. 基于SVM的中文微博情感分析的研究 [D]. 2011.
[3] B Liu. Sentiment analysis and opinion [J]. Synthesis Lectures on Human Language Technologies, 2012, 5 (1): 1-167.
[4] 徐冰. 基于统计学习的文本情感分析关键技术研究 [D]. 哈尔滨工业大学, 2012.
[5] 王钦炀, 施水才, 王洪俊. 文本情感分析综述 [J/OL]. [2024-11-19]. https://www.cnki.com.cn/Article/CJFDTotal-RJDK20241101003.htm.
[6] 燕道成, 沈鼎洲. “日本排放核污水”事件中央视新闻微博用户评论的文本情感分析 [J]. 传媒, 2024 (16): 88-90.
[7] 杜利明, 郭文艳, 崔蕾, 等. 基于LDA的电商平台用户评论挖掘与情感分析研究——以京东商城App为例 [J]. 江苏科技信息, 2024, 41 (12): 125-129.
[8] 周胜臣, 瞿文婷, 石英子, 等. 中文微博情感分析研究综述 [J]. 计算机应用与软件, 2013, 30 (3): 161-164, 181.
[9] 徐琳宏, 林鸿飞, 潘宇, 等. 情感词汇本体的构造 [J]. 情报学报, 2008, 27 (2): 180-185.
[10] 朱嫣岚, 闵锦, 周雅倩, 等. 基于HowNet 的词汇语义倾向计算 [J]. 中文信息学报, 2006, 20 (1): 14-20.
[11] 侯敏, 滕永林, 李雪燕, 等. 话题型微博语言特点及其情感分析策略研究 [J]. 语言文字应用, 2013 (2): 135-143.
[12] 孙建旺, 吕学强, 张雷瀚. 基于词典与机器学习的中文微博情感分析研究 [J]. 计算机应用与软件, 2014, 31 (7): 177-181.
[13] 党蕾, 张蕾. 一种基于知网的中文句子情感倾向判别方法 [J]. 计算机应用研究, 2010, 27 (4): 1370-1372.
[14] 赵妍妍, 秦兵, 车万翔, 等. 基于句法路径的情感评价单元识别 [J]. 软件学报, 2011, 22 (5): 887-898.
[15] J Wiebe. Learning subjective adjectives from corpora [C]. In Proceedings of AAAI-2000, 2000: 735-740.
[16] G E Hinton, P R Salakhutdinov. Reducing the dimensionality of data with neural network [J]. Science, 2006, 28 (7): 504-507.
[17] T Mikolov, M Karafiát, L Burget, et al. Recurrent neural network based language model [C] // INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, Sept. 26-30, 2010: 1045-1048.
[18] T Mikolov, K Chen, G Corrado, et al. Efficient estimation of word representations in vector space [C] // ICLR 2013. 1st International Conference on Learning Representations, Scottsdale, AA, USA, May 2-4, 2013: 1-12.
[19] J Pennington, R Socher, C D Manning. Glove: Global vectors for word representation [J]. Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP' 2014), 2014 (12): 1532-1543.
[20] I Sutskever, O Vinyals, Q V V Le. Sequence to sequence learning with neural networks [C]. Advances in Neural Information Processing Systems, 2014: 3104-3112.
[21] D Bahdanau, K Cho, Y Bengio. Neural machine translation by jointly learning to align and translate [C] // ICLR 2015. 3rd International Conference on Learning Representations, San Diego, CA, USA, May 7-9, 2015: 1-15.
[22] H Palangi, L Deng, Y L Shen, et al. Deep sentence embedding using long-short-term memory networks: analysis and application to information retrieval [J]. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2016, 24 (4): 694-707.
[23] Y Shen, X He, J Gao, et al. Learning semantic

representations using convolutional neural networks for web search [C]. Proceedings of the Companion Publication of the 23rd International Conference on World Wide Web Companion. International World Wide Web Conferences Steering Committee, 2014: 373–374.
[24] Y Bengio, O Delalleau. On the expressive power of deep architectures [C] // Algorithmic Learning Theory. Springer Berlin Heidelberg, 2011: 18–36.
[25] 孙志军, 薛磊, 许阳明, 等. 深度学习研究综述 [J]. 计算机应用研究, 2012, 29 (8): 2806–2810.
[26] Y LeCun, Y Y Bengio, G Hinton. Deep learning [J]. Nature, 2015, 521(7553): 436–444.
[27] 双锴. 计算机视觉 [M]. 北京: 北京邮电大学出版社, 2020.
[28] 唐聃, 白宁超, 冯暄, 等. 自然语言处理理论与实战 [M]. 北京: 电子工业出版社, 2018.
[29] 潘丽敏, 白崇有, 罗森林, 等. 基于注意力双层LSTM的长文本情感倾向性分析方法 [P]. 中国专利: CN108446275A. 2018.8.24.

Classification of Text Sentiment on Positive and Negative Tendencies Via Deep Learning-based Approaches

Deng Yilin Xu Shuwen Gao Xin

Jiangsu Normal University, Xuzhou

Abstract: This paper proposes a deep learning-based approach on text sentiment classification according to the kernel objective and current research status for text sentiment classification. By summarizing and evaluating conventional machine learning schemes, their limitations in dealing with complex emotional expressions are analyzed. A deep learning-based approach is adopted to construct a classification model on positive and negative sentiments for analyzing the data and testing microblog comments. This deep learning model is based on tasks for text sentiment classification towards positive and negative tendencies, which enables implementation of preprocessing, training, and testing massive text data. Experimental results indicate that when the batch size is within 1000 times, an average accuracy of 98% is reached in the test set, while precise classifications on more than 100,000 positive and negative text sentiment are achieved. Meanwhile, concluding remarks and outlooks are presented with respect to difficult problems on emotional text classification, i.e., ill-structured and ironic texts, coarse-grained sentiment analysis, lack of cultural awareness, dependence on data and annotations, as well as the limitations of word embeddings.

Key words: Text sentiment classification; Natural language processing; Deep learning; Sentiment tendency; Microblog

https://doi.org/10.35534/pss.0606155