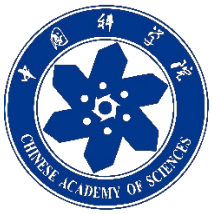


中国科学院大学

University of Chinese Academy of Sciences

文本数据挖掘大作业

成员 1：陈国鑫 2022E8013282125

成员 2：李一鸣 202228013229030

成员 3：张　兆 202228013229029

2023 年 06 月

目录

1 说明

在整个学期的课程中，三人共同学习，互相帮助，取长补短，完成课程作业期间，三人进行了充分的讨论，发挥了自身的优势，最终将成果汇总到这一篇报告中。**经过沟通，我们一致认为每个人对于本课程作业的贡献相同**。具体来说，每个人的侧重点稍稍不同，大概如下所示：

陈国鑫：实现了 TextCNN、LSTM 等经典深度学习模型及 BERT 微调等方法；部分参与文档的撰写工作。（约占比 33%）

李一鸣：使用 TF-IDF 抽取文本特征，并实现基于传统机器学习方法的情感分类方法；负责任务数据集的收集、清洗等工作；部分参与文档的撰写工作。（约占比 33%）

张兆：实现基于 BosonNLP 情感词典的情感分类方法；基于模型搭建文本情感识别 Web 系统并部署；部分参与文档的撰写工作。（约占比 33%）

我们将课程代码开源在 Github 上面供其他研究者交流与讨论，开源地址为：https://github.com/zhangzhao219/UCAS-2023-Spring-Homework/tree/Text-Data-Mining

2 引言

2.1 任务概览

情感分析是自然语言处理领域的一个重要分支，主要指通过对文本中的主观信息进行挖掘，判断文本作者的情感态度。

随着移动互联网的普及，网民已经习惯于在网络上表达意见和建议，比如电商网站上对商品的评价、社交媒体中对品牌、产品、政策的评价等等。这些评价中都蕴含着巨大的商业价值。比如某品牌公司可以分析社交媒体上广大民众对该品牌的评价，如果负面评价忽然增多，就可以快速采取相应的行动。而这种正负面评价的分析就是情感分析的主要应用场景。除此之外，应用场景还包括电影评论、舆情监控等。

文本情感分析指的是利用算法来分析提取文本中表达的情感。例如分析一个句子表达的好、中、坏等判断，高兴、悲伤、愤怒等情绪。如果能将这种文字转为情感的操作让计算机自动完成，就节省了大量的时间。对于目前的海量文本数据来说，这是很有必要的。

2.2 数据描述

为了全面评价情感识别的效果，我们选取了四种不同类型的数据集，包括IMDB 英文二分类数据集，Climate 英文多分类数据集，Waimai 中文二分类数据集和 IEMOCAP 多模态数据集。下面对这几个数据集进行简要的介绍。

2.2.1 IMDB

IMDB 数据集是一个高质量的影评数据集，共有 50000 条数据，其中 25000 条是正向的影评，25000 条是负向的影评，正向与负向样本数量平衡。我们采用 Kaggle 上面经过基本预处理的版本，其中训练集有 39723 条，验证集有 4998 条，测试集有 4995 条，比例大概为 8：1：1。

2.2.2 Climate

Climate 数据集汇总了 2015 年 4 月 27 日至 2018 年 2 月 21 日期间收集的与气候变化有关的推特文本，总共有 43943 条。每条推特文本由 3 名评审员独立标记，判断文本对于气候变化的情感趋向。每条推特文本都被标记为以下类别之一：

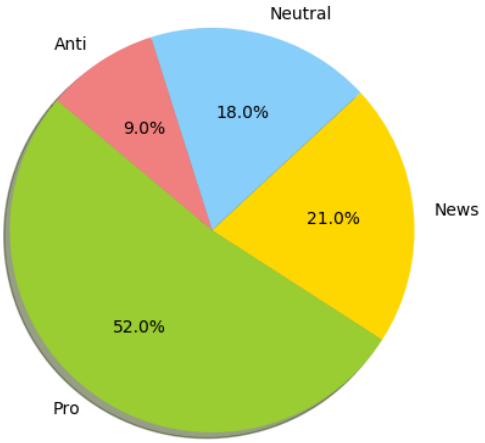
2：推特链接到有关气候变化的事实新闻

1：推特支持人为气候变化的观点

0：推特既不支持也不反驳人为气候变化的观点

-1：推特不相信人为的气候变化

样本类别并不平衡，类别比例分布如下图所示：



1 Climate 数据集标签分布情况

数据集没有经过划分，我们手动将数据集按照 7：2：1 划分训练集、验证集与测试集，每种集合的类别比例与整体数据集的比例大致相同。

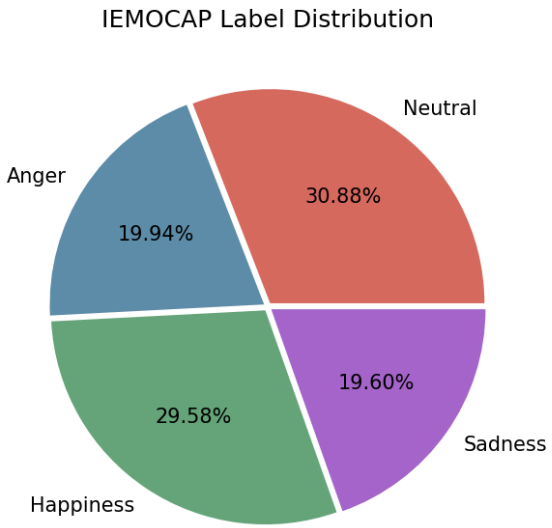
2.2.3 Waimai

Waimai 数据集是一个中文二分类数据集，来源于某平台用户对外卖的评价，只有正向评价和负向评价两种类别。其中总共有 11987 条数据，正向评论数目为 4000，负向评论数量为 7987。相同的，我们手动将数据集按照 7：2：1 划分训练集、验证集与测试集。

2.2.4 IEMOCAP

南加州（USC）语音分析和解释实验室（SAIL）收集了交互式情感二元运动捕捉数据库 IEMOCAP[2]，可以进行语音情感识别探究。该数据集包含大约十二个小时数据，记录了来自 10 位演员的二元会话信息，它们被要求在假设场景中即兴对话，旨在引发特定的情绪（快乐、愤怒、悲伤、沮丧和中性等状态），此外，数据集当中还包含演员的面部、手部动作等模态信息。

考虑到本文致力于探索文本情感识别，并考虑到平衡数据类别分布等因素，本文选择从数据库当中抽取音频信息转录的文本信息，由于某些类别存在较为严重的长尾现象，因此，本文选择了“neu”（中性）、“sad”（悲伤）、“hap”快乐、“ang”愤怒等四个类别，并将“exc”（激动）类别划分为“hap”类别。划分后的数据集包含 5531 条数据，数据分布如下图所示：



2 IEMOCAP 数据集标签分布情况

2.3 评价指标

本文选取四种评价标准对情感识别进行评价，在介绍评价标准的计算方式之前，首先要介绍四个概念，也被称为混淆矩阵。

True Positive（TP）：预测为正，实际为正

False Positive（FP）：预测为正，实际为负

True Negative（TN）：预测为负，实际为正

False Negative（FN）：预测为负，实际为负

（1）Acc.

准确率旨在知道总样本中预测对的概率，计算公式为：

$$Acc.=\frac{TP+TN}{TP+TN+FP+FN}$$

（2）Pre.

精确率又叫查准率，旨在预测为正的样本中时间为正的有多少。计算公式为：

$$Pre.=\frac{TP}{TP+FP}$$

（3）Rec.

召回率也叫查全率，旨在找到实际为正的样本中多少被预测为正。计算公式为：

$$Rec.=\frac{TP}{TP+FN}$$

（4）F1

F1 是为了既能体现精确率又能体现召回率的一个评价指标。计算公式为：

$$F1=\frac{2*Pre.*Rec.}{Pre.+Rec.}$$

对于多分类评价来说，我们采用 Macro 的权衡方式，即选取每一类的平均。

3 文本情感识别

3.1 数据处理

对于文本来说，与结构化的数值数据不同，首先要进行一些预处理操作，提高文本数据的完整性、正确性和一致性，从而提高模型训练和预测的效率。我们在将文本输入到传统机器学习模型之前进行了一系列的文本预处理操作：

（1）数据清洗。从原始的文本数据中删除标签为空或者标签不合理的数据，同时过滤一些明显长度过短的文本。

（2）分词。对于中文数据来说，文本一般是通过句子的形式给出，因此需要首先将句子划分为词语。在本次任务中我们使用 jieba 分词工具进行分词。Jieba

分词工具性能高效，准确性高，成熟稳定，是中文自然语言处理的重要基石之一。

（3）去除停用词。将一些经常出现在文本中但是语义不强的词汇，如“的”、“是”、“了”、“the”等进行去除，防止其数量过多稀释了重要的模型特征权重，影响模型性能。在本次任务中使用百度的停用词表进行过滤。

（4）保留某些标点符号。一般来说，对于文本的预处理是要移除掉所有的标点符号的，且标点符号也可以作为分词的一个依据。但是对于情感识别任务来说，问号、感叹号等标点符号在一定程度上反映了当前人的一些心理状态。因此需要保留这些比较特殊的标点符号作为文本的一种类别的特征。

3.2 基于情感词典的方法

基于情感词典的分析方法。是文本情感挖掘分析方法中的一种，其普遍做法是首先对文本进行情感词匹配，然后汇总情感词进行评分，最后得到文本的情感倾向。目前使用较多的情感词典是 BosonNLP 情感词典，其是由波森自然语言处理公司推出的一款已经做好标注的情感词典，词典中对每个情感词进行情感值评分。

基于 BosonNLP 情感词典的情感分析原理比较简单。对文本进行 jieba 分词后将分词好的列表数据对应 BosonNLP 词典进行逐个匹配，并记录匹配到的情感词分值，最后统计汇总所有情感分值。如果总分值大于 0，表示情感倾向为积极的；如果总分值小于 0，则表示情感倾向为消极的。

基于情感词典的方法可以准确反映文本的非结构化特征，易于分析和理解。在这种方法中，当情感词覆盖率和准确率高的情况下，情感分类效果比较准确。

但基于情感词典的情感分类方法主要依赖于情感词典的构建，由于现阶段网络的快速发展，信息更新速度的加快，出现了许多网络新词，对于许多类似于歇后语、成语或网络特殊用语等新词的的识别并不能有很好的效果，现有的情感词典需要不断地扩充才能满足需要；情感词典中的同一情感词可能在不同时间、不

同语言或不同领域中所表达的含义不同，因此基于情感词典的方法在跨领域和跨语言中的效果不是很理想；在使用情感词典进行情感分类时，往往考虑不到上下文之间的语义关系。

3.3 传统机器学习方法

除了之前进行的一系列的数据预处理对文本数据进行清洗外，还需要将文本的特征提取出来，构造模型可以识别的数值型输入，才可以使用机器学习模型进行情感识别。

将文本转换为数字特征的方式有很多，例如词袋模型，n-gram 模型，词嵌入等等。在本次任务的传统方法部分，我们使用了 TF-IDF 提取文本的特征。

TF-IDF 的主要思想是：如果一个词或短语在一篇文章中出现的频率高，同时在其他文章中很少出现，则认为此词或短语对该文章具有很强的相关性，具有很好的区分度。这体现了词语在语义上的重要性，是信息检索与文本挖掘中常用的评价指标之一。

TF-IDF 通过以下两个指标来衡量一个词语的重要性：

1. 词频（Term Frequency, TF）:某个词语在一篇文档中出现的次数。词频可以衡量一个词语在该文档中的重要程度，出现次数越多，该词可能越重要。

2. 逆文档频率（Inverse Document Frequency, IDF）：某个词语在整个文档集合中的词频，用来衡量该词语在所有文档中的常见程度。如果一个词在很多文档中出现，那么该词的区分度较小，IDF 值会较小。相反，如果一个词只在少数文档出现，该词的 IDF 值会较大，说明其具有很好的区分度。

TF-IDF 的值由这两个指标共同决定，通常使用以下公式计算：

$$TF - IDF = TF * IDF = (\textit{词频}) * log(\textit{总文档数}/\textit{包含该词的文档数})$$

其中，词频 TF 衡量词语在某个文档中的重要性，IDF 通过惩罚在许多文档

中出现的常用词，而提高只在少数文档中出现的词的权重，起到平衡的作用。

TF-IDF 可以有效地衡量一个词语的重要性。由于它考虑了词语在该文档和所有文档中的频率信息，所以可以过滤掉常见但语义较弱的词，更强调比较具有区分度的词语。这使得 TF-IDF 特别适用于文本的特征提取与关键词识别等任务。

3.3.1 线性回归

线性回归是一种最简单的机器学习算法，假设特征与标签之间存在线性关系。线性回归的模型表达为：

$$y = w_1x_1 + w_2x_2 + \ldots + w_nx_n + b$$

其中，x_1到x_n是n个特征，w_1到w_n是每个特征对应的权重，b是模型的偏置。对任意一个特征x_i，如果其权重w_i为正，那么x_i对y有正相关影响；如果w_i为负，那么x_i对y有负相关影响。权重的大小表示该特征对模型预测的重要性。对于本次的任务来说，特征即为通过 TF-IDF 提取的文本特征，标签即为各类别的实际标签。

线性回归旨在找到一组权重w和偏置b，可以使得模型对训练数据的预测与真实标签y^*尽量接近。这是一个优化问题，常用的损失函数是均方误差，通过梯度下降等方法不断调整w和b的值以最小化损失函数，从而得到最优的模型。

线性回归方法比较简单，容易理解，且计算效率高；缺点是假定了线性关系，因此无法处理非线性问题，学习能力有限，且容易过拟合，对异常值比较敏感。对于本次任务来说，由于标签是离散的值，因此只能将经过模型后输出的值与标签绝对距离进行比较，实际上线性回归并不适用于本次的任务。

3.3.2 Logistic 回归

逻辑回归是一种广泛使用的机器学习算法，用于解决二分类问题，也可以进行扩展从而解决多分类问题。

逻辑回归模型表达为：

$$y = sigmoid(w_1x_1 + w_2x_2 + \ldots + w_nx_n + b)$$

其中，x_1到x_n是 n 个特征，w_1到w_n是每个特征对应的权重，b是模型的偏置。

逻辑回归训练的目的是最大化训练数据被正确分类的概率。它采用最大似然估计法，通过梯度上升等优化方法不断更新权重w和b的值，使模型对训练数据的分类结果与真实标签y^*尽可能吻合。

多类逻辑回归模型的原理如下：

1. 假设类别之间是互斥的，即每个实例只能属于一个类别。对每个类别i,引入y_i表示该实例属于类别i的概率。

2. 为每个类别构建一个逻辑回归模型,预测属于该类别的概率。具体而言:

$$p(y_i = 1|x) = exp(w_i^T x)/(1 + exp(w_i^T x))$$

这里w_i是类别i对应的权重向量。

3. 对新输入x，在每个逻辑回归模型中计算属于对应类别的概率，然后选择概率最大的类别作为预测类别。

多类逻辑回归的优点在于原理简单，易于理解和实现，并且可以在不加修改的情况下，扩展到更多类别。缺点是无法学习类别之间的相关性，且计算量比较大，训练代价高。

3.3.3 朴素贝叶斯

朴素贝叶斯是一种简单实用的机器学习算法，用于解决分类问题。原理如下：

1. 假设特征之间相互独立。这是朴素贝叶斯算法的核心假设，该假设简化了模型使其易于计算。

2. 根据贝叶斯定理计算后验概率。对于某个类别C和特征向量X，贝叶斯定理为：

$$P(C|X) = P(X|C)P(C) / P(X)$$

其中$P(C|X)$是后验概率，$P(X|C)$是联合概率，$P(C)$是先验概率

3. 构建分类器。对于新输入x，计算它属于每个类别的后验概率，然后预测类别为先验最大的那个类别。

朴素贝叶斯模型表达为：

$$P(C|x_1, x_2, \ldots, x_n) = P(C) * P(x_1|C) * P(x_2|C) * \ldots * P(x_n|C)/P(x_1, x_2, \ldots, x_n)$$

其中x_1到x_n是n个特征，$P(C)$是类别C的先验概率，$P(x_i|C)$是特征x_i在类别C下的条件概率。

朴素贝叶斯的“朴素”是因为它假定特征相互独立。这个假设简化了模型,使其易计算,但也限制了其表达能力。然而即便特征之间存在依赖,朴素贝叶斯仍然表现良好。这是因为贝叶斯定理在分类效果上更为重要。朴素贝叶斯通过训练统计特征-类别共现频次，获得先验概率和条件概率。预测阶段直接应用贝叶斯定理,以获得每个类别的后验概率。

朴素贝叶斯的优点是计算简单，易理解且易实现。缺点是其对属性独立的假设限制了模型的表达能力，且对小训练集表现不佳。

3.3.4 决策树

决策树是一种流行的机器学习算法，用于解决分类和回归问题。其原理如下：

1. 以树形结构表示决策过程。决策树由根节点、内部节点(非叶子节点)和叶子节点组成。

2. 根节点代表完整训练数据集。内部节点表示数据的某个特征，每个子节点代表该特征的一个取值。

3. 叶子节点代表数据集的分类结果或回归结果。分类树的叶子节点包含类别标签，回归树的叶子节点包含连续值。

4. 决策树通过递归地对数据进行切分，直到达到停止条件，逐步生成。

5. 对新数据实例，从根节点开始，依据其特征取值递归地向下移动，最终到达叶子节点，获得预测结果。

决策树主要学习算法有：ID3、C4.5、CART。其中 CART 构建决策树通过贪心算法，按照特征选择和数据切分的方式生成二叉树。其具体步骤为：

1. 选择最优特征和切分点：计算各个特征和各个切分点的代价，选择代价最小的特征和切分点进行切分。

2. 数据切分：将节点的数据切分成子节点，根据最优切分点的取值将数据划分到左子节点或右子节点。

3. 生成子节点：为左右子节点重复以上步骤，直到达到停止条件。

4. 剪枝：采用 CART 算法可以剪去一些子节点，防止决策树模型过拟合。

决策树算法结果易于理解，对异常值不太敏感。但可能导致过拟合，学习结果随训练数据的变化而变化大。决策树是最简单,使用最广的机器学习算法之一。

3.3.5 随机森林

随机森林是一种流行的机器学习算法，用于解决分类和回归问题。它的原理如下：

1. 随机森林由多个决策树组成。每个决策树独立进行训练，最后将多个决策树的结果综合得到最终输出。

2. 随机森林使用 bagging 的思想，通过有放回地从训练数据中抽样获得子数据集来训练每个决策树。

3. 构建每个决策树时，随机选取特征子集。这增加了随机森林的多样性，避免过拟合。

4. 对新输入实例，将其输入到所有的决策树，并统计各类结果出现的次数，最终预测出现次数最多的类。

由于数据、特征的随机抽取，每棵决策树不同，随机森林具有较强的多样性，从而减少了过拟合的风险。随机森林准确性高、对异常值鲁棒、易于理解和实现。缺点是计算开销大、训练时间长。

3.3.6 支持向量机

支持向量机（SVM）是一种用于解决分类和回归问题的机器学习算法。它的原理如下：

1. SVM 通过寻找超平面将不同类的训练数据分开，并且使得该超平面与最

近的训练数据点之间的距离尽可能大。

2. SVM 使用核技巧隐式地将数据映射到高维空间，使数据在高维空间线性可分，而在原始空间不可分。常用有线性核、多项式核和 RBF 核等。

3. SVM 使用最大间隔分类器，寻找能将不同类的训练数据分开的超平面，使得该超平面与最近的数据点的距离最大。

4. SVM 因此得到一个分类器，能够对新数据判断类别。对于非线性可分数据集，SVM 可以通过核技巧映射得到其在高维空间的表现，仍然使用最大间隔分类器构建超平面，从而实现非线性分类。

SVM 理论基础坚实、泛化能力强，通过核技巧可处理非线性问题，对中间空间数据不敏感。缺点是核的选择和超参数的调优比较困难。

3.4 深度学习方法

3.4.1 TextCNN & BiLSTM

文本情感分类是一项重要的自然语言处理任务，常常需要利用深度学习技术进行处理。在这些深度学习技术中，TextCNN[9]和双向长短期记忆网络（BiLSTM）[10]已被证明可以有效地应用于文本情感分类任务。

TextCNN：TextCNN 是一种具有多通道输入、多尺度卷积和全局最大池化等特征的卷积神经网络模型。它能够充分利用文本的局部和全局信息，并且能够高效地捕捉文本中的关键特征。因此，在情感分类任务中，TextCNN 能够有效地提高分类准确率和鲁棒性，并且具有良好的效率和可扩展性。TextCNN 的网络架构示意图如下：

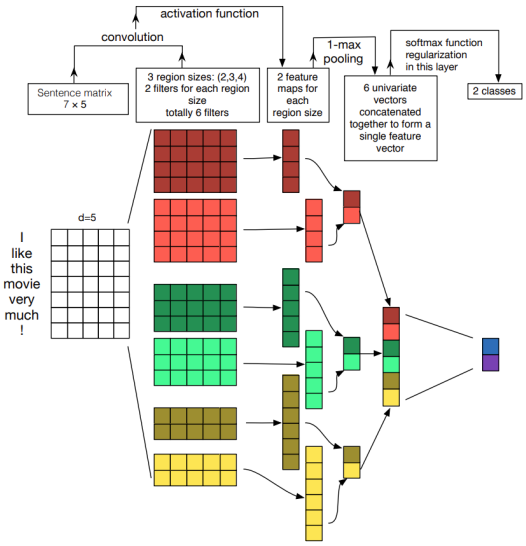


图 3 TextCNN 模型示意图

BiLSTM：BiLSTM 是一种基于循环神经网络（RNN）的模型，在处理时序数据上表现出色。BiLSTM 能够通过正向和反向传递信息，充分利用文本的上下文信息，并且具有优秀的建模能力。在情感分类任务中，BiLSTM 能够更好地处理长文本序列，并且对上下文信息的利用更加精细和全面，因此可以取得更好的分类效果。BiLSTM 的网络架构示意图如下：

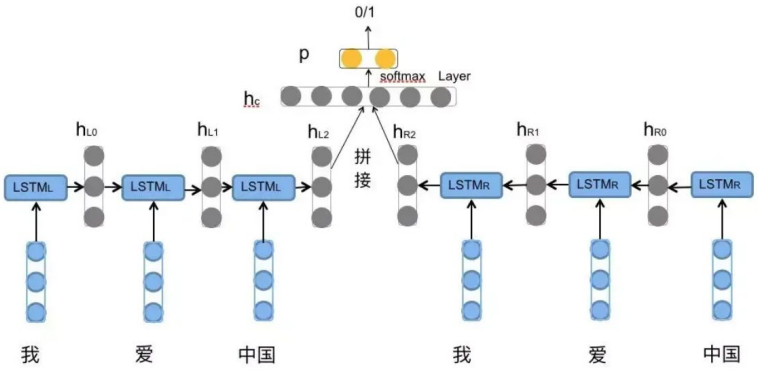


图 4 BiLSTM 模型示意图

通过预训练的词向量模型得到文本数据的词向量矩阵，在本文中包含中文和英文两种语言的数据集，分别采用“tencent-ailab-embedding-zh-d200”和“glove.42B.300d”作为预训练词向量模型。

3.4.2 预训练模型

预训练模型是指在大规模无标签语料库上进行预训练，学习到通用的语言知识和表征能力，然后针对具体的下游任务（如文本分类、命名实体识别等），进行有监督的微调。

Bert[11]是一种基于 Transformer[12]架构的预训练语言模型，被广泛应用于自然语言处理任务中。在情感分类任务中，Bert 可以利用大规模语料库的上下文信息和语义特征，并且能够针对不同类型的情感分类任务进行微调，从而取得较好的分类效果。具体来说，Bert 首先使用无监督的方式进行预训练，通过掩码语言模型和下一句预测等任务，学习到了丰富的语言知识和表征能力。在情感分类任务中，Bert 能够充分利用文本的上下文和语义信息，对于长文本和复杂情感分类任务有很好的应用效果。此外，还可以通过多层抽取和特征融合等技术，进一步提高分类效果。Bert 的网络架构示意图如下：

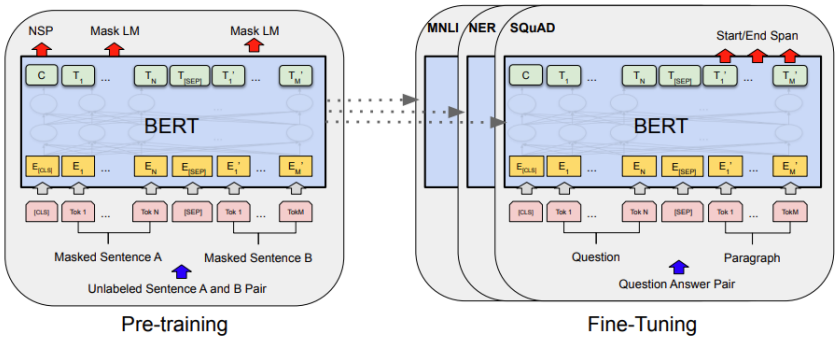


图 5 Bert 模型示意图

在 Bert 中，每一个文本序列都会被添加特殊标记 [CLS] 和 [SEP]，其中 [CLS] 标记表示整个文本序列的“类别”。在句子级别的情感分类任务中，可以将文本序列作为输入，同时在序列的开头添加 [CLS] 标记，并将该位置的输出向量作为整个句子的表示向量。在本文实验中，我们尝试对不同规模的 Bert 进行微调，在多个数据集上测试其效果。

3.4.3 Prompt Tuning

Prompt 技术[13]是一种基于语言模型的自然语言处理技术，它通过在输入文本中插入提示语（Prompt），来引导模型生成与指定任务相关的输出。Prompt 技术

可以帮助解决少样本学习、零样本学习等问题，并且具有较好的可解释性和易用性。其中，Soft Prompt[14]是一种特殊的 Prompt 技术，它采用软提示方式对模型进行微调。与硬提示不同，Soft Prompt 不会强制指定任务的确切格式和内容，而是使用一组关键词或短语来引导模型生成相关的输出。具体来说，在训练时，Soft Prompt 可以将预定义的提示词或短语加入到输入序列中，作为模型学习的一部分。在推理时，模型将这些提示词或短语视为一种信号或信息，从而更好地完成特定的任务。Soft Prompt 的主要优点是其灵活性和可扩展性。相比硬提示，它能够更准确地匹配输入数据的多样性，从而提高模型的泛化能力。其中较为经典的 Soft Prompt 方法为 P-Tuning v2[15]。与之前的 Soft Prompt 方法相比，P-Tuning v2 引入了多层 Prompt 的机制，同时采用了强化学习算法来优化 Prompt 的生成，提高了模型的适应性和泛化能力。P-Tuning v2 的模型示意图如下：

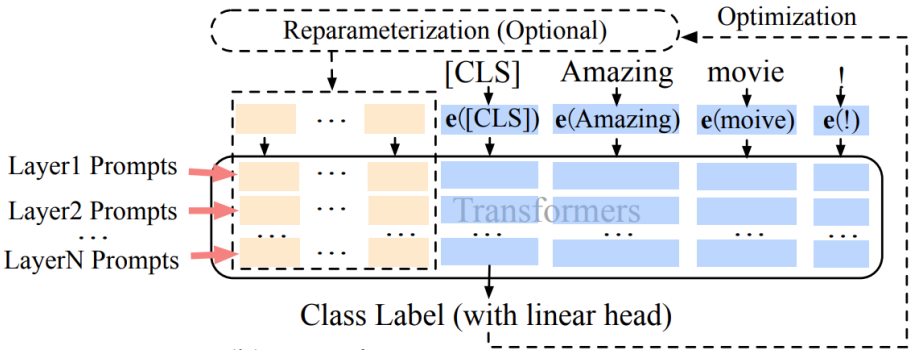


图 6 P-Tuning v2 的模型示意图

在文本实验中，我们设置 Prompt token 数为 20，并且采用多层感知机（MLP）进行重参数化，提高模型效果。

3.5 文本情感分析实验

表 1 在 IMDB 和 Climate 数据集上进行情感识别的实验结果

Model	IMDB				Climate			
	Acc.	Pre.	Rec.	F1	Acc	Pre.	Rec.	F1
Dict	0.501	0.501	0.988	0.665	/	/	/	/
Linear	0.8238	0.8246	0.8238	0.8237	0.5522	0.5451	0.3885	0.3813

Logistic Regression	0.8288	0.8292	0.8288	0.8287	0.6680	0.6253	0.5292	0.5549
Naïve Bayes	0.7910	0.7912	0.7910	0.7909	0.6351	0.5852	0.4831	0.5000
Decision Tree	0.6966	0.6967	0.6966	0.6966	0.5659	0.4769	0.4743	0.4754
Random Forest	0.7784	0.7797	0.7783	0.7781	0.6200	0.6611	0.4758	0.4859
SVM	0.8282	0.8286	0.8282	0.8281	0.6604	0.6223	0.5035	0.5256
TextCNN	0.9000	0.9000	0.9000	0.9000	0.7449	0.7001	0.6835	0.6903
BiLSTM	0.9006	0.9009	0.9006	0.9006	0.7454	0.7029	0.6937	0.6972
bert-base	0.9360	0.9373	0.9360	0.9360	0.7899	0.7540	0.7381	0.7446
bert-large	0.9412	0.9412	0.9412	0.9412	0.7909	0.7532	0.7426	0.7472
base-P-Tuning v2	0.8904	0.8905	0.8904	0.8904	0.7689	0.7346	0.6890	0.7067
large-P-Tuning v2	0.9038	0.9039	0.9038	0.9038	0.7692	0.7272	0.7112	0.7172

表 2 在 Waimai 和 IEMOCAP 数据集上进行情感识别的实验结果

Model	WaiMai				IEMOCAP			
	Acc.	Pre.	Rec.	F1	Acc.	Pre.	Rec.	F1
Dict	0.6546	0.4903	0.9083	0.6368	/	/	/	/
Linear	0.7787	0.8040	0.6875	0.7038	0.4512	0.5530	0.4231	0.4036
Logistic Regression	0.7741	0.8067	0.6778	0.6926	0.6282	0.6265	0.6368	0.6303
Naïve Bayes	0.7824	0.8161	0.6896	0.7066	0.5884	0.6042	0.5749	0.5861
Decision Tree	0.7713	0.7848	0.6819	0.6969	0.5144	0.5037	0.5189	0.5095
Random Forest	0.7731	0.7939	0.6813	0.6964	0.5090	0.5994	0.4825	0.4690
SVM	0.7750	0.8006	0.6819	0.6974	0.6119	0.6102	0.6201	0.6138
TextCNN	0.8685	0.8527	0.8507	0.8517	0.6264	0.6386	0.6312	0.6338
BiLSTM	0.8778	0.8650	0.8576	0.8611	0.6336	0.6322	0.6477	0.6389
bert-base	0.9065	0.8935	0.8972	0.8953	0.7058	0.7034	0.7251	0.7117
bert-large	/	/	/	/	0.7238	0.7144	0.7295	0.7207
base-P-Tuning v2	0.9046	0.8966	0.8868	0.8914	0.6841	0.6727	0.7042	0.6843
large-P-Tuning v2	/	/	/	/	0.7238	0.7099	0.7362	0.7203

从测试结果可以看出，基于情感词典的分类效果并不好，可能原因是 BosonNLP 情感词典与本次选取的数据集匹配程度不高，尤其是 IMDB 数据集，基于情感词典的方法的召回率将近 1，说明通过情感词典计算出来的所有文本的情感倾向基本相同。这种结果显然是非常不可靠的。

在传统机器学习方法中，Logistic Regression 和 SVM 比较适合，相较于其他的机器学习方法的性能较高，其中 Logistic Regression 对大规模数据有很好的扩展性，SVM 的模型泛化能力强。

深度学习方法可以达到更高的识别精度，最高的精度与机器学习方法相比要高出 0.1 左右，这是因为深度学习模型可以自动学习文本的语义特征表示，而不

再依赖手工提取的特征。相比之下，传统机器方法需要人工选取特征，依赖于人工经验和专业知识。

但是，深度学习模型在其他方面也有不足之处，深度学习模型通常具有较高的模型复杂度，需要大数据集进行训练，计算资源要求也更高，而传统机器学习模型相对简单，计算资源需求较小，且具有更好的解释性。

综上所述，深度学习方法在文本情感识别任务上具有更高的识别精度和更好的泛化能力等优势。但其模型较为复杂，需要大量数据和计算资源。相比之下，传统机器学习方法更加简单和可解释，但识别效果有限，且需要专业知识进行特征工程。

4 文本情感识别系统

在提升算法性能的同时，为了方便查看端到端的情感识别效果，我们实现了一个简单的情感识别系统，支持用户在本地自行输入文本，然后在网页上实时查看情感识别的效果。系统的后端使用 Python 的 Flask 框架，前端使用简单的 HTML+CSS+Javascript 进行搭建。

在后端调用我们训练好的模型进行推理的时候，其中的文本模型我们使用 Mindspore 框架进行重新部署。Mindspore 框架相对于 Pytorch 框架来说，对 CPU 推理的支持更好，部署简单且性能优异。部署流程使用 Docker 进行自动化控制，部署的服务器为华为云 2 核 4G 的云耀云服务器，硬盘大小为 40G。

系统的效果展示如下：

图 7 文本情感识别系统界面示意图

左侧为文本的输入框，输入文字后点击上传文本，然后点击下方按钮即可以在右侧展示情感分类的结果。其中的粉色圆形内部是文本情感分类模型推理的数值形式的标签，粉色文字是推理的实际标签。可以看出，对于输入的文本来说，推理效果正确。

5 总结与展望

从本次实验的研究结果来看，基于情感词典的方法的性能与情感词典有很大关系。基于传统机器学习的情感分类方法主要在于情感特征的提取以及分类器的组合选择，不同分类器的组合选择对情感分析的结果有存在一定的影响，这类方法在对文本内容进行情感分析时常常不能充分利用上下文文本的语境信息，存在忽略上下文语义的问题，因此对其分类准确性会有一定的影响。深度学习使用语言模型预训练的方法充分利用了大规模的单语语料，可以对一词多义进行建模，使用语言模型预训练的过程可以被看作是一个句子级别的上下文词表示。通过对大规模语料预训练，使用一个统一的模型或者将特征加到一些简单的模型中，在很多 NLP 任务中取得了不错的效果，说明这种方法在缓解对模型结构的依赖问

题上有明显的效果。

基于文本的情感识别有如下的几个发展方向：

（1）通过对比不同的研究方法可以发现，现有的对于情感分析的研究方法多基于单一领域，如社交网络媒体平台 weibo、酒店评论等，在个性化推荐中如何将多个领域的内容结合，进行情感分类，实现更好的推荐效果，并实现在提高模型的泛用性能，都是未来值得研究和探索的工作方向。

（2）大部分对于情感分析的研究多用于显式的文本情感分类问题，采用含有明显情感词的数据集，而对于某些隐式词的检测和分类效果不佳。现阶段对于隐式情感分析的研究还处于起步阶段，不是很充分，未来可以通过构建隐式情感词词典，或者通过使用更好的深度学习方法来更深层次地提取语义相关信息来实现更好的情感分类效果。

（3）对于复杂语句的情感分析研究需要进一步完善，当带有情感倾向的网络用语、歇后语、成语等越来越频繁地出现，尤其在文本中含有反讽或隐喻类的词时，情感极性的检测就会存在难度，这也需要进一步研究。

（4）多模态情感分析也是近来的研究热点，如何将多个模态中的情感信息进行提取和融合，是大家主要研究的方向，当多个模态中的情感表达不一致时，该如何权重不同模态中的情感信息也是需要考虑的；以及是否能考虑外部语义信息，这对情感分析的准确性是否有帮助，也是需要有大量的研究。

（5）预训练模型是现阶段的研究热点，它能有效解决传统方法中存在的问题，如不能并行化计算的限制等，还能有效捕获词语之间的相互关系，并且通过微调就能在下游任务中实现较好的效果，但也会存在模型参数量大，训练时间较长的问题。如何在模型的参数量小，有效缩短训练时间的前提下，达到好的分类效果，也会是值得研究的方向。

6 参考文献

[1] Devlin J, Chang M W, Lee K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding[J]. arXiv preprint arXiv:1810.04805, 2018.

[2] Sohn K, Berthelot D, Carlini N, et al. Fixmatch: Simplifying semi-supervised

learning with consistency and confidence[J]. Advances in neural information processing systems, 2020, 33: 596-608.

[3] Sun Y, Wang S, Feng S, et al. Ernie 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation[J]. arXiv preprint arXiv:2107.02137, 2021.

[4] Miyato T, Dai A M, Goodfellow I. Adversarial training methods for semi-supervised text classification[J]. arXiv preprint arXiv:1605.07725, 2016.

[5] Wei C, Sohn K, Mellina C, et al. Crest: A class-rebalancing self-training framework for imbalanced semi-supervised learning[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021: 10857-10866.

[6] Wei J, Zou K. EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks[C]//Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing. 2019: 6382-6388.

[7] Alammar, J (2018). The Illustrated Transformer [Blog post]. Retrieved from https://jalammar.github.io/illustrated-transformer/

[8] Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What Does BERT Learn about the Structure of Language?. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

[9] Chen Y. Convolutional neural network for sentence classification[D]. University of Waterloo, 2015.

[10] Huang Z, Xu W, Yu K. Bidirectional LSTM-CRF models for sequence tagging[J]. arXiv preprint arXiv:1508.01991, 2015.

[11] Devlin J, Chang M W, Lee K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding[J]. arXiv preprint arXiv:1810.04805, 2018.

[12] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. Advances in neural information processing systems, 2017, 30.

[13] Liu P, Yuan W, Fu J, et al. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing[J]. ACM Computing Surveys, 2023, 55(9): 1-35.

[14] Lester B, Al-Rfou R, Constant N. The power of scale for parameter-efficient prompt tuning[J]. arXiv preprint arXiv:2104.08691, 2021.

[15] Liu X, Ji K, Fu Y, et al. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks[J]. arXiv preprint arXiv:2110.07602, 2021.