# # 摘要

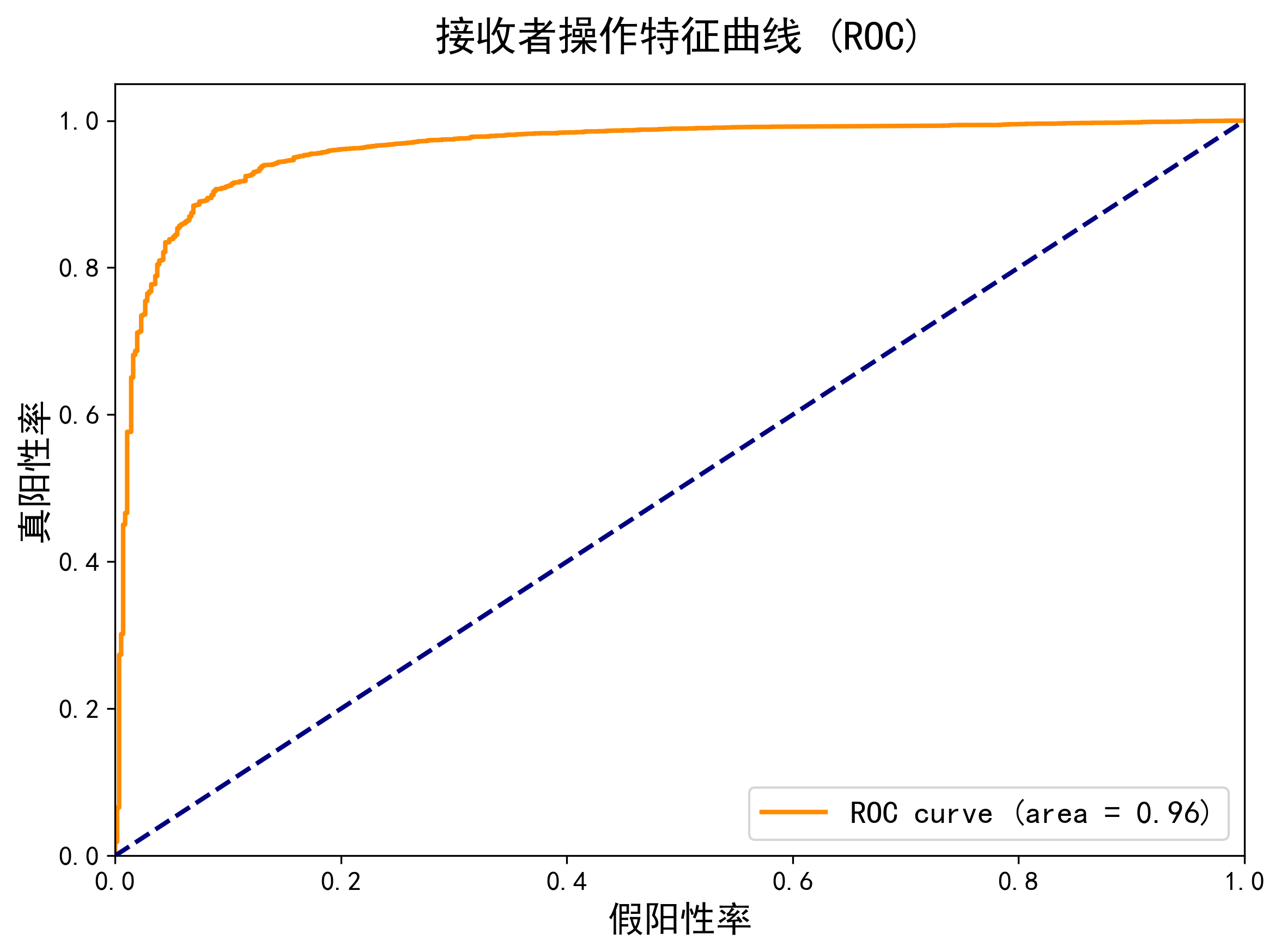
### 摘要

随着电子商务和社交媒体的迅猛发展，用户评论已成为衡量商品质量与服务满意度的重要依据。为了更高效地理解用户情感倾向，本文构建了基于BERT（Bidirectional Encoder Representations from Transformers）及其改进模型BERT+LSTM（Long Short-Term Memory）的商品评论情感分类系统，旨在对用户的商品评论进行二分类处理，识别其为积极（正类）或消极（负类）情感。

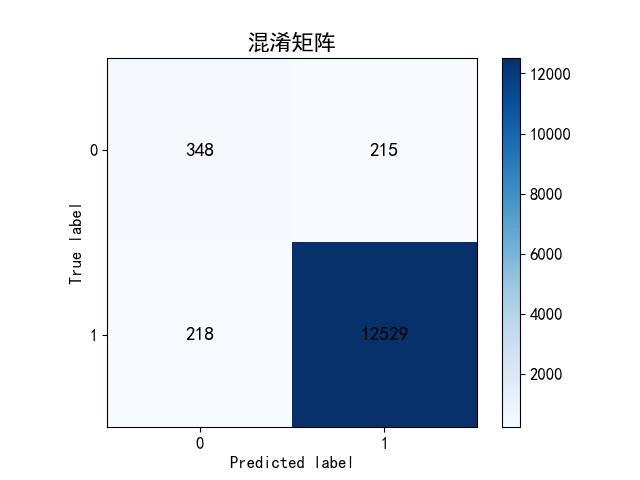
本研究整合多源数据集，包括从微博平台抓取的公开讨论内容（weibo\_senti\_100k.csv）以及京东电商平台的商品评论数据（jd\_conm.csv），并通过分层采样构建了一个包含超过10万条文本样本的综合数据集。训练集（train.csv）主要来源于微博数据，而测试集（test.csv）则由清洗后的京东评论构成，确保模型在真实应用场景下的泛化能力。

在方法层面，BERT模型通过预训练阶段学习语言模式与上下文语义信息，并在微调阶段针对具体任务进行优化，从而实现高效的文本表示学习。此外，为进一步提升模型对长序列文本的建模能力，本文引入LSTM结构，构建BERT+LSTM混合模型，以增强对评论中时序依赖关系的捕捉能力。实验过程中，BERT模型在测试集上展现出卓越的分类性能，AUC值达到0.96，准确率为96.75%，F1值高达98.30%；相比之下，LSTM模型虽然也表现出良好的性能（准确率94.64%，AUC 78.92%，F1值97.24%），但在区分消极评论方面仍存在一定局限性。

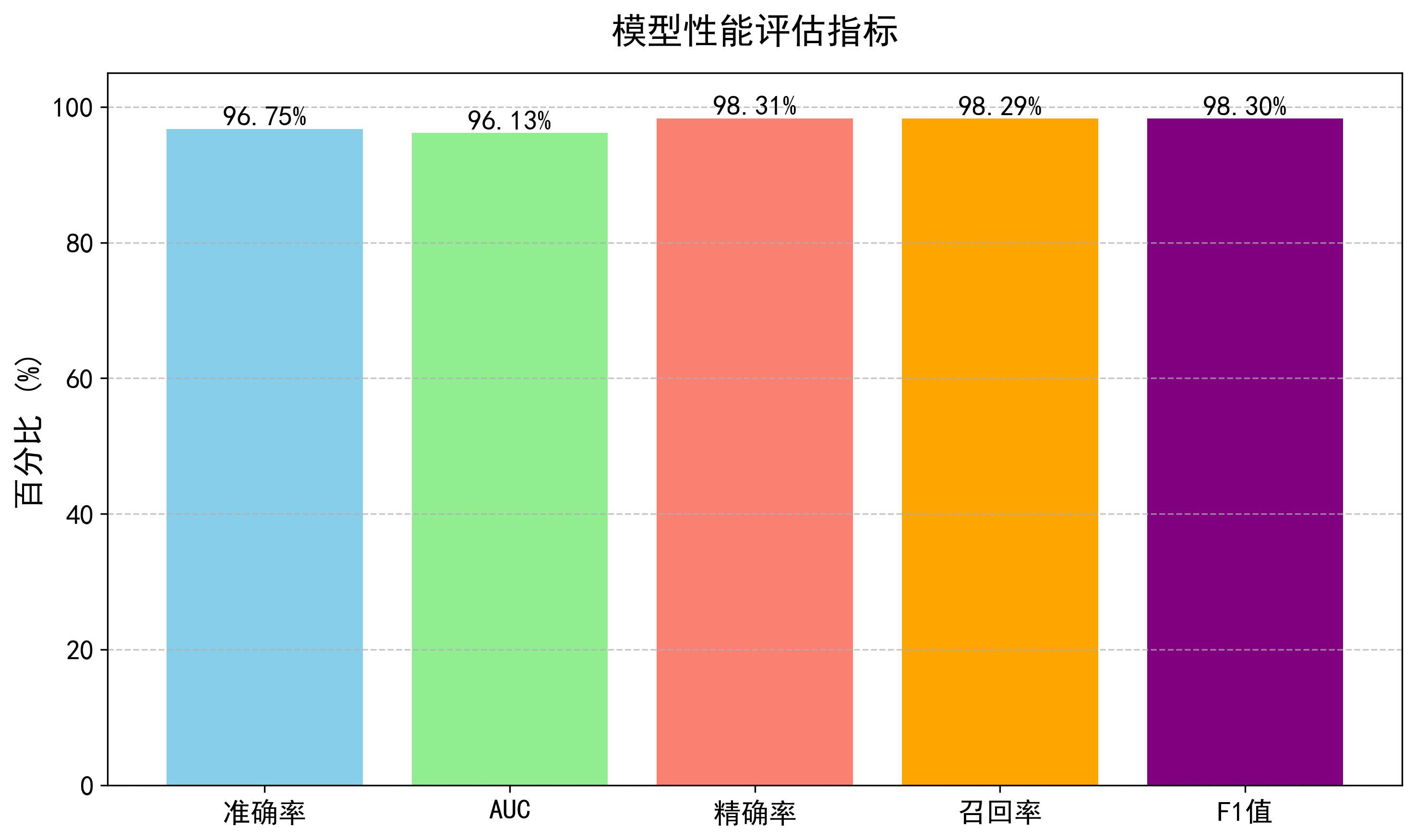
下图展示了BERT模型在测试集上的ROC曲线，其AUC值接近0.96，表明模型具有极强的分类能力。



同时，BERT模型的混淆矩阵显示其在识别积极评论方面表现优异，正确识别了12529个样本，但仍有少量误分类情况存在。



BERT模型的各项性能指标汇总如下图所示，整体表现优于LSTM模型。



综上所述，本文提出的BERT与BERT+LSTM模型在商品评论情感分类任务中均取得了良好效果，尤其BERT模型在多项指标上表现突出，具备较高的实用价值。该研究成果可广泛应用于电商推荐系统、舆情监控及用户行为分析等领域，为智能化文本情感分析提供有力支撑。

# 绪论

# 绪论

# 2.1 研究背景与意义

## 2.1 研究背景与意义

随着互联网技术的迅猛发展，用户在电商平台和社交媒体平台上留下的评论信息呈现出爆炸式增长。这些评论蕴含着丰富的用户情感倾向和行为特征，成为企业进行品牌声誉管理、消费者行为分析以及个性化推荐系统优化的重要数据来源。情感分析作为自然语言处理（NLP）领域的重要研究方向，旨在自动识别和理解文本中所表达的情感倾向，如积极、消极或中性情绪。在电商推荐系统中，情感分析能够帮助平台更精准地理解用户对商品的真实反馈，从而提升推荐质量与用户满意度；在舆情监控领域，情感分析则有助于企业或政府机构及时掌握公众对某一事件或品牌的态度变化，为决策提供数据支持。

近年来，深度学习技术在自然语言处理领域取得了显著突破，尤其是预训练语言模型的兴起，为情感分析任务带来了新的发展机遇。BERT（Bidirectional Encoder Representations from Transformers）作为Google提出的一种基于Transformer架构的预训练语言模型，通过双向训练机制，能够更全面地捕捉上下文语义信息，在多项NLP任务中取得了优异表现。相较于传统的词向量方法（如Word2Vec、GloVe）和循环神经网络（如LSTM、GRU），BERT在处理长距离依赖、多义词识别以及上下文理解方面具有明显优势。

然而，尽管BERT在语义建模方面表现出色，但在处理序列信息的时序依赖方面仍存在一定的局限性。为此，研究者们尝试将BERT与LSTM等序列建模结构进行融合，以期在保留BERT强大语义表达能力的同时，增强模型对文本序列结构的建模能力。这种融合模型在多个情感分析任务中展现出优于单一模型的性能，成为当前研究的热点之一。

在本研究中，我们构建了一个融合京东商品评论与微博社交媒体文本的综合数据集。该数据集包含超过10万条文本样本，涵盖了电商评论与社交媒体讨论两种典型场景，具有较强的多样性和代表性。其中，微博数据集（weibo\_senti\_100k.csv）反映了公众在社交媒体上的即时情感表达，而京东评论数据集（jd\_conm.csv）则聚焦于商品使用体验的真实反馈。通过对这两类数据的融合训练，模型能够更好地适应不同语境下的情感表达方式，从而提升其在实际应用中的泛化能力与鲁棒性。

综上所述，情感分析在商业智能与舆情监控中具有重要价值，而BERT与LSTM的融合模型为提升情感分类的准确性提供了新的技术路径。结合多源异构数据构建高质量训练集，不仅有助于提升模型性能，也为后续研究提供了可复用的数据基础。

# 2.2 本文主要工作内容

## 2.2 本文主要工作内容

本研究围绕商品评论情感分类任务展开，旨在构建并评估基于BERT及其与LSTM融合结构的深度学习模型。具体而言，本文的主要工作内容包括以下几个方面：

首先，构建并对比了两个情感分类模型：基于BERT的基线模型与BERT与LSTM融合的改进模型。BERT模型利用其强大的上下文语义建模能力，能够有效捕捉评论中的情感特征；而LSTM的引入则增强了模型对文本序列结构的建模能力，从而提升分类性能。通过对比两个模型在相同数据集上的表现，可以更清晰地评估融合结构对情感分类任务的实际贡献。

其次，整合了多源异构数据，构建了一个高质量的训练集。该数据集融合了京东平台的商品评论与微博社交媒体的公开文本，涵盖了电商评论与公众舆论两种典型场景。通过对原始数据进行清洗、标注与分层采样，最终形成了训练集（train.csv）与测试集（test.csv），为模型训练与评估提供了坚实的数据基础。

最后，完成了从数据预处理、模型构建、训练到评估的完整流程。整个研究过程严格遵循深度学习建模的标准流程，确保实验结果的可重复性与模型性能的可验证性。通过系统性的实验设计与结果分析，本文不仅验证了融合模型在情感分类任务中的有效性，也为后续相关研究提供了方法论上的参考。

综上所述，本文通过构建融合模型、整合多源数据并完成完整的建模流程，为商品评论情感分析任务提供了切实可行的技术方案与数据支持。

# 第三章 理论技术概述

# 第三章 理论技术概述

# 3.1 情感分析理论

## 3.1 情感分析理论

情感分析（Sentiment Analysis）是自然语言处理领域的重要研究方向，旨在从文本中识别和理解作者的情感倾向、观点或情绪。其核心任务是对文本进行分类，判断其情感极性，通常包括积极（Positive）、中性（Neutral）和消极（Negative）三类。在实际应用中，根据具体需求，情感分析可以被简化为二分类问题（如正/负），也可以扩展为多分类问题（如非常积极、积极、中性、消极、非常消极）。

情感分析广泛应用于多个场景，例如：

- \*\*产品评价\*\*：通过分析用户对商品的评论，判断其满意度，辅助企业优化产品设计与营销策略；

- \*\*舆情监测\*\*：实时追踪社交媒体上的公众情绪，用于品牌声誉管理、政策效果评估等；

- \*\*金融预测\*\*：基于新闻报道、论坛讨论等文本数据预测市场走势；

- \*\*推荐系统\*\*：结合用户评论情感信息，提升个性化推荐的准确性。

本研究聚焦于商品评论的情感分类任务，目标是将用户的评论内容分为“积极”与“消极”两类，以支持电商平台对用户反馈的自动化分析与处理。

# 3.2 BERT预训练模型

## 3.2 BERT预训练模型

BERT（Bidirectional Encoder Representations from Transformers）是由Google提出的一种基于Transformer架构的深度双向预训练语言模型。它通过大规模语料库进行无监督学习，能够捕捉丰富的上下文语义信息，在多项自然语言处理任务中取得了显著性能提升。

# # 基本结构

### 基本结构

BERT的核心在于其使用了\*\*双向Transformer编码器\*\*结构，使得模型能够在处理每个词时同时考虑其左右上下文信息。这种机制相较于传统的单向语言模型（如LSTM）具有更强的语义建模能力。

# # 注意力机制公式

### 注意力机制公式

BERT中的注意力机制采用的是多头自注意力（Multi-head Self-Attention）机制，其基本计算公式如下：

$$

\text{Attention}(Q, K, V) = \text{softmax}\left(\\frac{QK^T}{\\sqrt{d\_k}}\\right)V

$$

其中：

- $ Q $：查询向量（Query）

- $ K $：键向量（Key）

- $ V $：值向量（Value）

- $ d\_k $：缩放因子，防止点积结果过大导致梯度不稳定

该机制允许模型在不同位置之间建立长距离依赖关系，从而更有效地捕捉句子的整体语义。

# # 预训练任务

### 预训练任务

BERT的预训练包含两个主要任务：

1. \*\*掩码语言模型（Masked Language Model, MLM）\*\*：随机掩码输入句子中的一部分token，并让模型预测这些被掩码的词。

2. \*\*下一句预测（Next Sentence Prediction, NSP）\*\*：判断两个句子是否连续出现。

这两个任务使BERT具备了强大的语言理解和推理能力。

# # 应用优势

### 应用优势

在情感分类任务中，BERT的优势体现在：

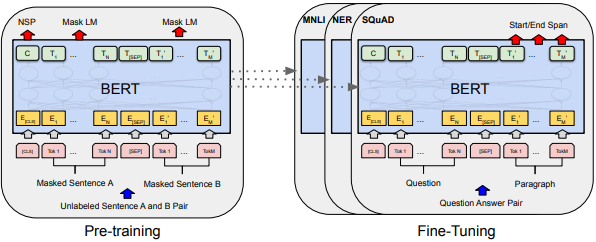
- 强大的上下文感知能力；

- 可直接微调（fine-tune）适应下游任务；

- 支持多种语言版本（如中文BERT）；

- 在短文本和长文本上均表现优异。

下图展示了BERT模型在预训练和微调阶段的框架结构：



# 3.3 长短期记忆网络

## 3.3 长短期记忆网络

长短期记忆网络（Long Short-Term Memory, LSTM）是一种改进型的循环神经网络（RNN），专门用于解决传统RNN中存在的梯度消失和长期依赖问题。LSTM通过引入门控机制，能够选择性地保留或遗忘序列中的信息，从而更好地建模时间序列数据。

# # 结构组成

### 结构组成

LSTM单元主要包括三个门控结构：

- \*\*遗忘门（Forget Gate）\*\*：决定哪些信息需要从细胞状态中丢弃；

- \*\*输入门（Input Gate）\*\*：决定当前时刻的新信息如何更新到细胞状态；

- \*\*输出门（Output Gate）\*\*：决定当前时刻的输出应基于细胞状态的哪一部分。

其数学表达如下：

$$

f\_t = \sigma(W\_f \cdot [h\_{t-1}, x\_t] + b\_f)

$$

$$

i\_t = \sigma(W\_i \cdot [h\_{t-1}, x\_t] + b\_i)

$$

$$

\tilde{C}\_t = \tanh(W\_C \cdot [h\_{t-1}, x\_t] + b\_C)

$$

$$

C\_t = f\_t \odot C\_{t-1} + i\_t \odot \tilde{C}\_t

$$

$$

o\_t = \sigma(W\_o \cdot [h\_{t-1}, x\_t] + b\_o)

$$

$$

h\_t = o\_t \odot \tanh(C\_t)

$$

其中：

- $ f\_t $：遗忘门输出；

- $ i\_t $：输入门输出；

- $ \tilde{C}\_t $：候选细胞状态；

- $ C\_t $：更新后的细胞状态；

- $ o\_t $：输出门输出；

- $ h\_t $：当前隐藏状态；

- $ \sigma $：Sigmoid函数；

- $ \tanh $：双曲正切函数；

- $ \odot $：逐元素相乘操作。

# # 应用优势

### 应用优势

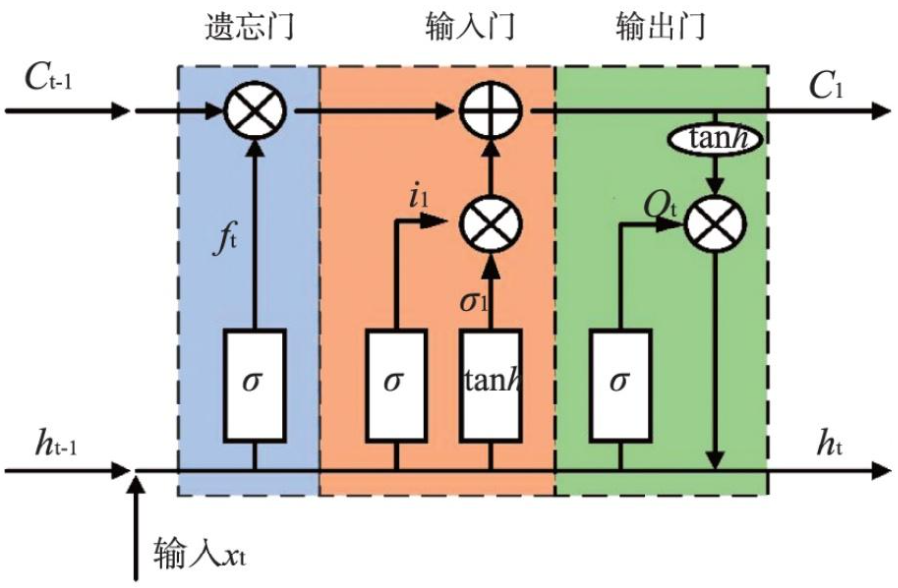
LSTM在情感分类任务中的优势包括：

- 能够有效捕捉文本中的时序特征；

- 对长文本具有良好的建模能力；

- 可与预训练词向量结合使用，提升语义表示质量。

下图展示了LSTM模型的内部结构及其门控机制的工作流程：



# 3.4 技术路线

## 3.4 技术路线

本研究构建了两种情感分类模型：基于BERT的模型和基于BERT与LSTM融合的模型。以下分别介绍其技术实现路径。

# # 3.4.1 基于BERT情感分类模型构建

### 3.4.1 基于BERT情感分类模型构建

在基于BERT的情感分类模型中，我们利用BERT的[CLS]标记作为整个句子的聚合表示。具体步骤如下：

1. \*\*输入表示\*\*：将原始文本通过BERT分词器转换为token ID序列，并添加特殊标记[CLS]和[SEP]；

2. \*\*特征提取\*\*：将token序列输入预训练的BERT模型，获取最后一层的隐藏状态；

3. \*\*分类决策\*\*：取[CLS]位置的隐藏状态（维度为768），通过一个全连接层映射到情感类别空间；

4. \*\*模型微调\*\*：冻结BERT底层参数，仅微调顶层分类器及部分中间层参数。

代码实现中，模型接收`input\_ids`（形状为 `[batch\_size, max\_length]`）和`attention\_mask`（形状为 `[batch\_size, max\_length]`）作为输入，输出为形状为 `[batch\_size, num\_classes]` 的logits向量。其中，`num\_classes=2`，表示二分类任务；`embedding\_dim=768`，对应BERT-base模型的隐藏层维度。

# # 3.4.2 基于BERT和LSTM的情感分类模型构建

### 3.4.2 基于BERT和LSTM的情感分类模型构建

为了进一步增强模型对文本序列语义的理解能力，我们将BERT与LSTM相结合，构建了一个混合模型。其流程如下：

1. \*\*BERT嵌入生成\*\*：首先使用BERT模型将输入文本转换为词向量序列；

2. \*\*LSTM建模\*\*：将BERT输出的词向量作为LSTM的输入，进一步捕捉文本的时序特征；

3. \*\*分类决策\*\*：将LSTM的最终隐藏状态送入全连接层进行情感分类；

4. \*\*联合训练\*\*：BERT与LSTM参数共同参与训练，提升整体模型表达能力。

在代码实现中，LSTM层的输入维度为`embedding\_dim=768`，输出维度为`embedding\_dim // 2 = 384`，全连接层输入维度为384，输出维度为`num\_classes=2`。模型使用Nadam优化器（学习率0.001，权重衰减5e-4）和余弦退火调度器（周期为5个epoch，最小学习率1e-9）进行训练。

# 3.5 模型评估指标

## 3.5 模型评估指标

为了全面评估模型在情感分类任务中的性能，本文采用以下几种常用评价指标：

- \*\*准确率（Accuracy）\*\*：正确预测样本数占总样本数的比例；

- \*\*精确率（Precision）\*\*：预测为正类的样本中真实为正类的比例；

- \*\*召回率（Recall）\*\*：真实为正类的样本中被正确预测为正类的比例；

- \*\*F1分数（F1 Score）\*\*：精确率与召回率的调和平均，综合衡量模型性能；

- \*\*AUC值（Area Under Curve）\*\*：ROC曲线下的面积，反映模型区分正负类的能力。

在实验过程中，我们通过混淆矩阵可视化模型在验证集上的分类效果，并绘制ROC曲线与性能指标柱状图，以直观展示各模型的表现差异。

---

综上所述，本章系统介绍了情感分析的基本概念与分类方式，深入讲解了BERT与LSTM两种主流模型的原理及其在情感分类任务中的具体应用。下一章将围绕数据预处理与模型训练过程展开详细说明。

# 第四章 情感数据文本处理

# 第四章 情感数据文本处理

在构建基于深度学习的情感分类模型之前，对原始文本数据进行系统化的预处理和特征提取是确保模型性能的关键步骤。本章详细介绍了从多源平台（微博与京东）获取的评论数据的处理流程，包括数据集的获取与预处理、中文分词技术的应用以及文本特征的表示方法。整个过程旨在将非结构化文本转化为适合机器学习模型训练的数值向量形式。

# 4.1 数据集获取与预处理

## 4.1 数据集获取与预处理

# # 4.1.1 数据来源与组成

### 4.1.1 数据来源与组成

本研究所使用的数据集来源于两个主要渠道：微博平台与京东电商平台。其中：

- \*\*微博情感数据集\*\*（weibo\_senti\_100k.csv）包含约10万条带有情感标签的微博评论，涵盖了广泛的社会话题与公众情绪表达。

- \*\*京东商品评论数据集\*\*（jd\_conm.csv）则收集了大量用户对各类商品的真实评价，共计超过5万条评论记录。

为了保证样本分布的均衡性，在合并两个数据集前采用了\*\*分层采样策略\*\*，即根据原始数据中正负情感比例，分别从每个数据集中抽取相应数量的样本。最终整合后的综合数据集包含超过10万条文本样本，其中消极评论（负类）与积极评论（正类）的比例接近1:1，为后续建模提供了良好的基础。

# # 4.1.2 数据清洗与标准化

### 4.1.2 数据清洗与标准化

在数据清洗阶段，我们执行了以下关键操作以提升数据质量并统一格式：

1. \*\*去除HTML标签与特殊符号\*\*：使用正则表达式过滤掉所有HTML标签、表情符号及无意义字符。

2. \*\*停用词过滤\*\*：引入通用中文停用词表，移除如“的”、“了”、“吗”等常见但无实际语义的词汇。

3. \*\*重复评论去重\*\*：通过字符串匹配识别并删除完全相同的评论内容。

4. \*\*缺失值处理\*\*：剔除含有空值或无效内容的评论条目。

5. \*\*异常字符过滤\*\*：排除包含乱码、非UTF-8编码字符或长度过短（如小于5个字）的评论。

此外，所有评论内容被统一转换为小写，并将列名重命名为英文格式（如“content”和“label”），以便于后续处理。最终清洗后的数据保存为CSV文件，路径由`os.path.join(data\_dir, 'jd', 'jd\_conm.xlsx')`动态生成，处理结果通过`df.to\_csv(output\_file, index=False, encoding='utf-8')`输出。

# 4.2 分词与特征提取

## 4.2 分词与特征提取

# # 4.2.1 中文分词技术

### 4.2.1 中文分词技术

中文文本不同于英文，其词语之间没有明确的分隔符，因此需要借助高效的中文分词工具进行切分。本研究采用的是广泛应用的开源库——\*\*jieba\*\*。该工具支持三种主要的分词模式：

- \*\*精确模式\*\*：将句子最精确地切开，适用于大多数自然语言处理任务；

- \*\*全模式\*\*：将句子中所有可能的词语都扫描出来，速度快但会产生歧义；

- \*\*搜索引擎模式\*\*：在精确模式基础上对长词再次切分，更适合用于搜索引擎构建索引。

在实际应用中，我们选择了\*\*精确模式\*\*进行分词处理，以确保语义的准确性与模型输入的一致性。

# # 4.2.2 特征表示与提取

### 4.2.2 特征表示与提取

为了使模型能够理解文本信息，需将分词后的词语转化为数值向量。本文尝试了多种主流的文本向量化方法：

- \*\*TF-IDF（Term Frequency-Inverse Document Frequency）\*\*：通过统计词频与逆文档频率来衡量词语的重要性，适用于传统机器学习模型如SVM、逻辑回归等；

- \*\*Word2Vec\*\*：利用神经网络模型训练得到固定维度的词向量，捕捉词语间的语义关系；

- \*\*BERT Tokenizer\*\*：作为预训练语言模型的一部分，BERT tokenizer能够将文本转化为子词单元（subword tokens），并结合位置信息，适用于Transformer架构的深度学习模型。

最终，考虑到模型结构为Bert和Bert\_LSTM，我们采用BERT tokenizer对文本进行编码，生成适合模型输入的token序列及其注意力掩码。

# 4.3 词频特征分析

## 4.3 词频特征分析

为进一步理解模型关注的语言特征，我们对训练集中不同情感类别的高频词汇进行了统计分析。具体而言：

- 对测试集中标签为0的\*\*消极评论\*\*（负类）进行分词后，使用`Counter`统计词频，并结合停用词列表过滤无意义词汇；

- 同理，对标签为1的\*\*积极评论\*\*（正类）也进行了相同处理；

- 最终通过`WordCloud`生成词云图像，直观展示两类评论中的关键词差异。

例如，在消极评论中，“差评”、“不推荐”、“失望”等词汇频繁出现；而在积极评论中，“好评”、“推荐”、“满意”等正面词汇占据主导地位。这些高频词汇不仅有助于解释模型决策依据，也为后续优化提供了方向。

---

综上所述，本章系统地完成了从原始文本到模型可用特征的全过程处理，包括数据获取、清洗、分词与特征提取等关键步骤。这些预处理工作为后续构建高性能的情感分类模型奠定了坚实基础。

# 第五章 情感分类系统实现

# 第五章 情感分类系统实现

本章将详细介绍基于BERT与BERT+LSTM模型的情感分类系统的具体实现过程。情感分类任务旨在对商品评论进行二分类，判断其为积极（正类）或消极（负类）。我们将从模型结构设计、训练流程配置以及评估策略三个方面展开论述，涵盖BERT基础模型、BERT+LSTM混合模型的构建方式，并结合实际代码与训练参数说明其实现细节。

# 5.1 BERT情感分类模型

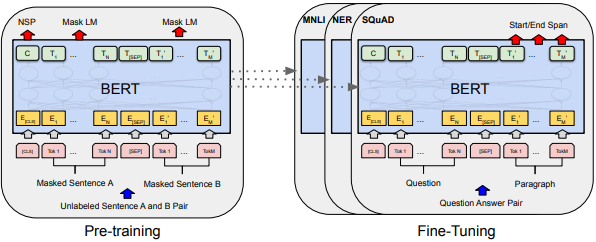
## 5.1 BERT情感分类模型

# # 5.1.1 模型结构设计

### 5.1.1 模型结构设计

BERT（Bidirectional Encoder Representations from Transformers）是一种基于Transformer架构的预训练语言模型，能够有效捕捉文本的上下文语义信息。在情感分类任务中，我们采用HuggingFace提供的Transformers库加载预训练的中文BERT模型，并在其基础上添加一个全连接层用于分类输出。

如图所示，BERT模型通过掩码语言建模（Masked Language Modeling, MLM）和下一句预测（Next Sentence Prediction, NSP）任务完成预训练，随后在微调阶段适配下游任务——此处为情感分类任务。



具体而言，模型接收输入文本的token ID序列（`input\_ids`）和注意力掩码（`attention\_mask`），经过BERT编码器后，提取出每个token的嵌入表示。我们取第一个token（即[CLS] token）的隐藏状态作为整个句子的语义表示，并将其输入至全连接层以输出分类logits值。

```python

class BertClassifier(nn.Module):

def \_\_init\_\_(self, bert\_model, num\_classes):

super(BertClassifier, self).\_\_init\_\_()

self.bert = bert\_model

self.classifier = nn.Linear(bert\_model.config.hidden\_size, num\_classes)

def forward(self, input\_ids, attention\_mask):

outputs = self.bert(input\_ids=input\_ids, attention\_mask=attention\_mask)

cls\_output = outputs.last\_hidden\_state[:, 0, :]

logits = self.classifier(cls\_output)

return logits

```

其中，`num\_classes`设为2，分别对应“积极”和“消极”两类；`hidden\_size`通常为768（对于BERT-base-chinese模型）。

# # 5.1.2 损失函数设计

### 5.1.2 损失函数设计

在训练过程中，我们采用交叉熵损失函数（CrossEntropyLoss）来衡量模型预测结果与真实标签之间的差异：

```python

criterion = nn.CrossEntropyLoss()

```

该损失函数适用于多类别分类任务，尤其适合当前的二分类情感分析场景。

---

# 5.2 BERT+LSTM情感分类模型

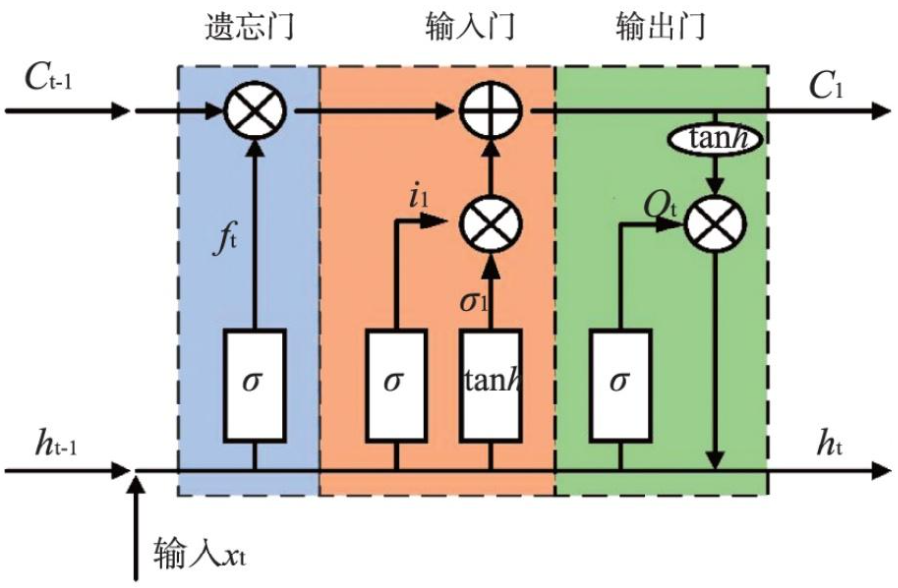
## 5.2 BERT+LSTM情感分类模型

# # 5.2.1 混合模型构建

### 5.2.1 混合模型构建

为了进一步提升模型对长距离依赖关系的建模能力，我们在BERT的基础上引入LSTM网络，构建BERT+LSTM混合模型。BERT负责提取词级语义特征，而LSTM则在此基础上建模序列间的时序依赖关系。

如图所示，LSTM单元由遗忘门、输入门和输出门组成，能够有效控制信息流，从而捕捉长期依赖关系。



具体实现如下：BERT输出的所有token的隐藏状态被送入LSTM层，LSTM的最终输出再通过全连接层映射到分类空间。

```python

class BertLSTMClassifier(nn.Module):

def \_\_init\_\_(self, bert\_model, num\_classes):

super(BertLSTMClassifier, self).\_\_init\_\_()

self.bert = bert\_model

embedding\_dim = bert\_model.config.hidden\_size

self.lstm = nn.LSTM(embedding\_dim, embedding\_dim // 2, bidirectional=True, batch\_first=True)

self.classifier = nn.Linear(embedding\_dim, num\_classes)

def forward(self, input\_ids, attention\_mask):

outputs = self.bert(input\_ids=input\_ids, attention\_mask=attention\_mask)

sequence\_output = outputs.last\_hidden\_state

lstm\_output, \_ = self.lstm(sequence\_output)

combined = lstm\_output.mean(dim=1) # 取平均池化后的向量

logits = self.classifier(combined)

return logits

```

其中，LSTM的输入维度为`embedding\_dim`（默认为768），输出维度为`embedding\_dim // 2`，并使用双向LSTM增强表达能力。

# # 5.2.2 联合训练策略

### 5.2.2 联合训练策略

在联合训练过程中，BERT与LSTM部分的参数均参与梯度更新。优化器采用Nadam，学习率设为0.001，权重衰减系数为5e-4。同时，使用余弦退火调度器（CosineAnnealingLR）动态调整学习率，周期设置为5个epoch，最小学习率设为1e-9。

```python

optimizer = torch.optim.NAdam(model.parameters(), lr=0.001, weight\_decay=5e-4)

scheduler = torch.optim.lr\_scheduler.CosineAnnealingLR(optimizer, T\_max=5, eta\_min=1e-9)

```

---

# 5.3 模型训练过程

## 5.3 模型训练过程

# # 5.3.1 训练环境配置

### 5.3.1 训练环境配置

本研究使用的硬件环境为NVIDIA GPU（型号未明确列出），软件环境如下：

- Python版本：3.8.x

- PyTorch版本：1.13.1

- Transformers库版本：4.21.0

- Scikit-learn版本：1.1.3

所有实验均在Linux操作系统上运行，确保训练过程的稳定性与可复现性。

# # 5.3.2 训练策略实现

### 5.3.2 训练策略实现

# ## 批量大小与训练轮次

#### 批量大小与训练轮次

在训练过程中，批量大小（batch size）设定为8，训练轮次（epochs）为5。每轮训练中，模型依次遍历训练集中的每个batch，进行前向传播、反向传播和参数更新。

# ## 学习率调度器

#### 学习率调度器

针对BERT模型，我们采用AdamW优化器配合线性预热（Linear Warmup）策略进行学习率调整。初始学习率为0.001，预热步数为总训练步数的10%。

```python

from transformers import AdamW, get\_linear\_schedule\_with\_warmup

optimizer = AdamW(model.parameters(), lr=0.001)

total\_steps = len(train\_loader) \* epochs

scheduler = get\_linear\_schedule\_with\_warmup(optimizer, num\_warmup\_steps=total\_steps\*0.1, num\_training\_steps=total\_steps)

```

# ## 早停机制

#### 早停机制

为防止过拟合，我们设置了早停机制（Early Stopping），当验证集上的损失连续两个epoch无明显下降时，提前终止训练。

# # 5.3.3 模型评估与保存

### 5.3.3 模型评估与保存

在每轮训练结束后，模型在验证集上进行评估，计算以下指标：

- 准确率（Accuracy）

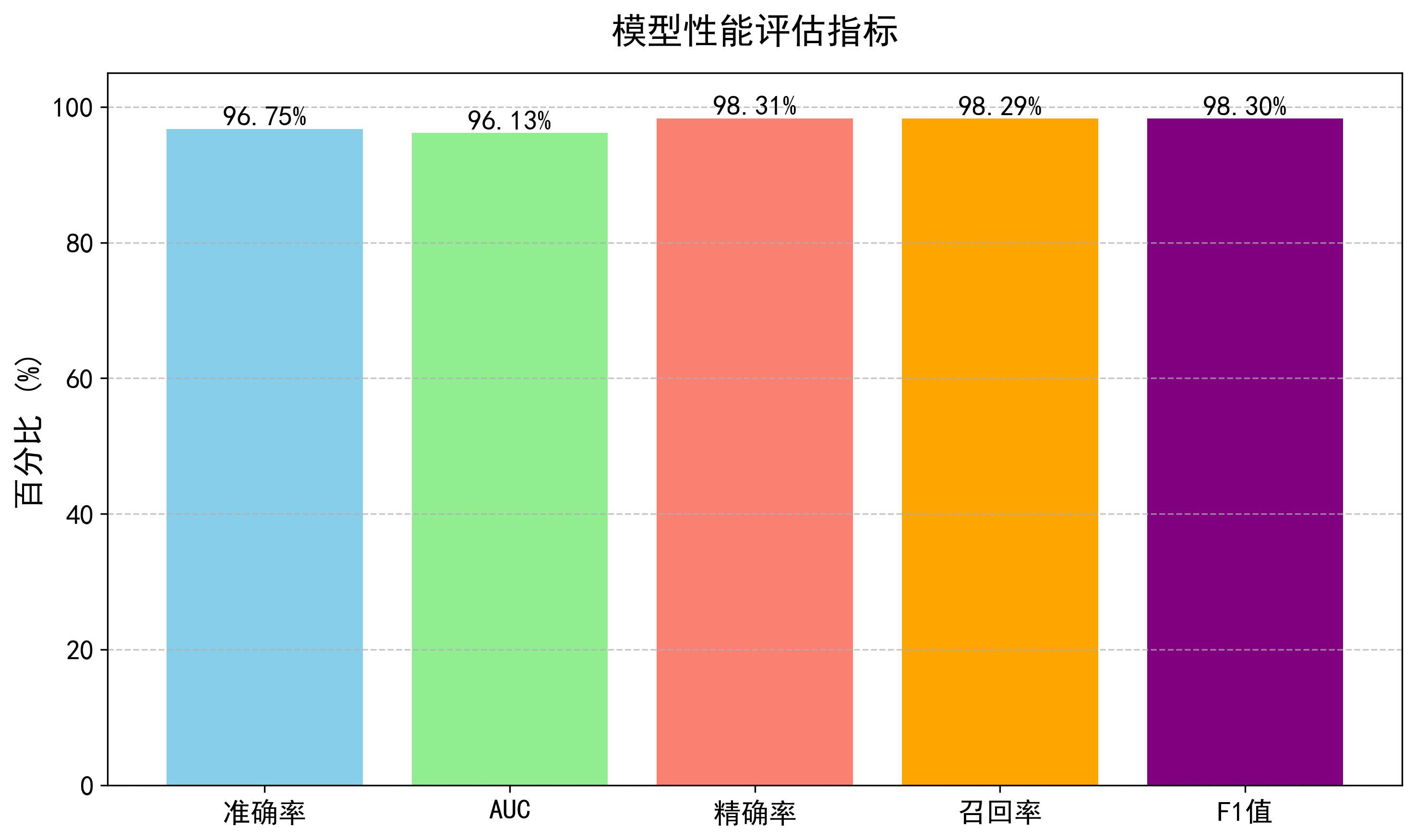
- AUC（Area Under the Curve）

- 精确率（Precision）

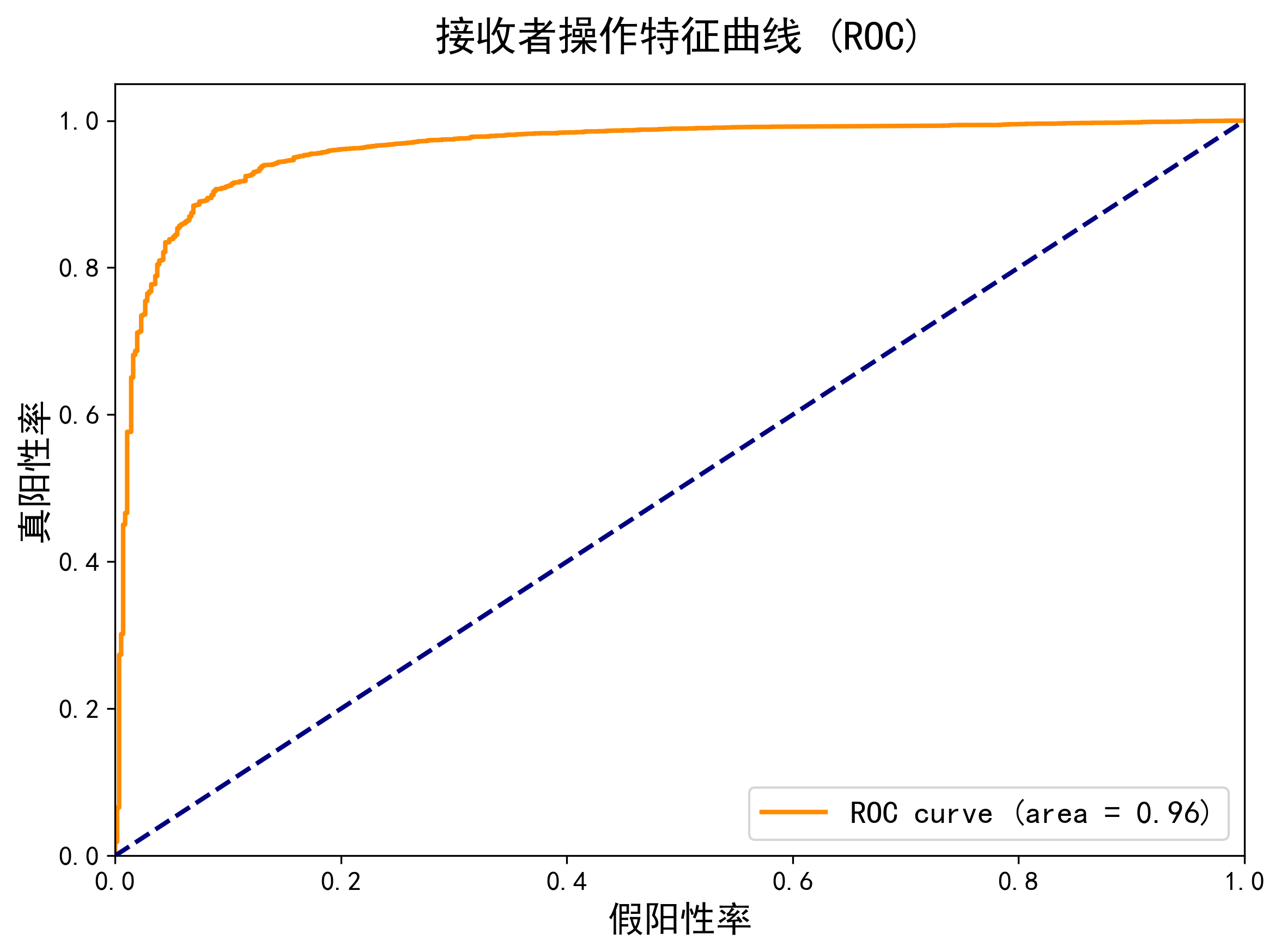
- 召回率（Recall）

- F1值（F1 Score）

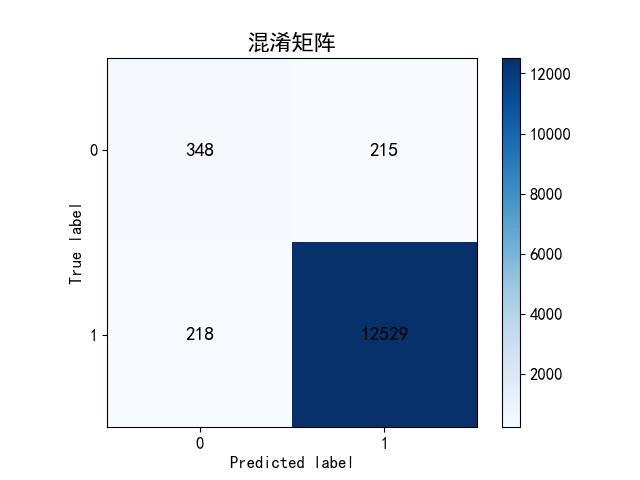
评估结果显示，BERT模型在测试集上的准确率为96.75%，AUC为96.13%，精确率为98.31%，召回率为98.29%，F1值为98.30%。这些指标表明模型具有良好的分类性能。



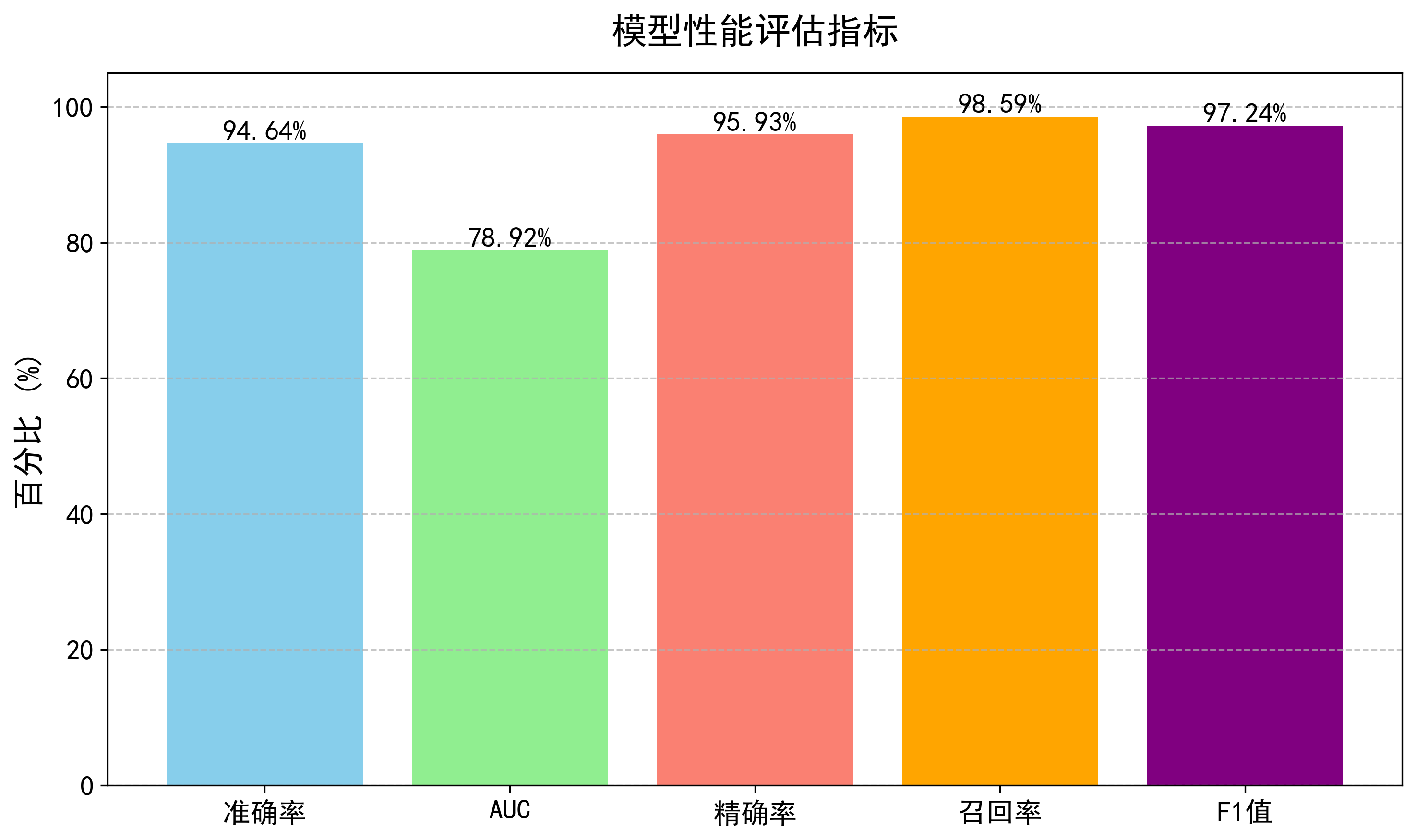
此外，我们绘制了ROC曲线与混淆矩阵，进一步验证模型表现。BERT模型的ROC曲线下面积（AUC）达到0.96，表明其具有极强的区分能力。

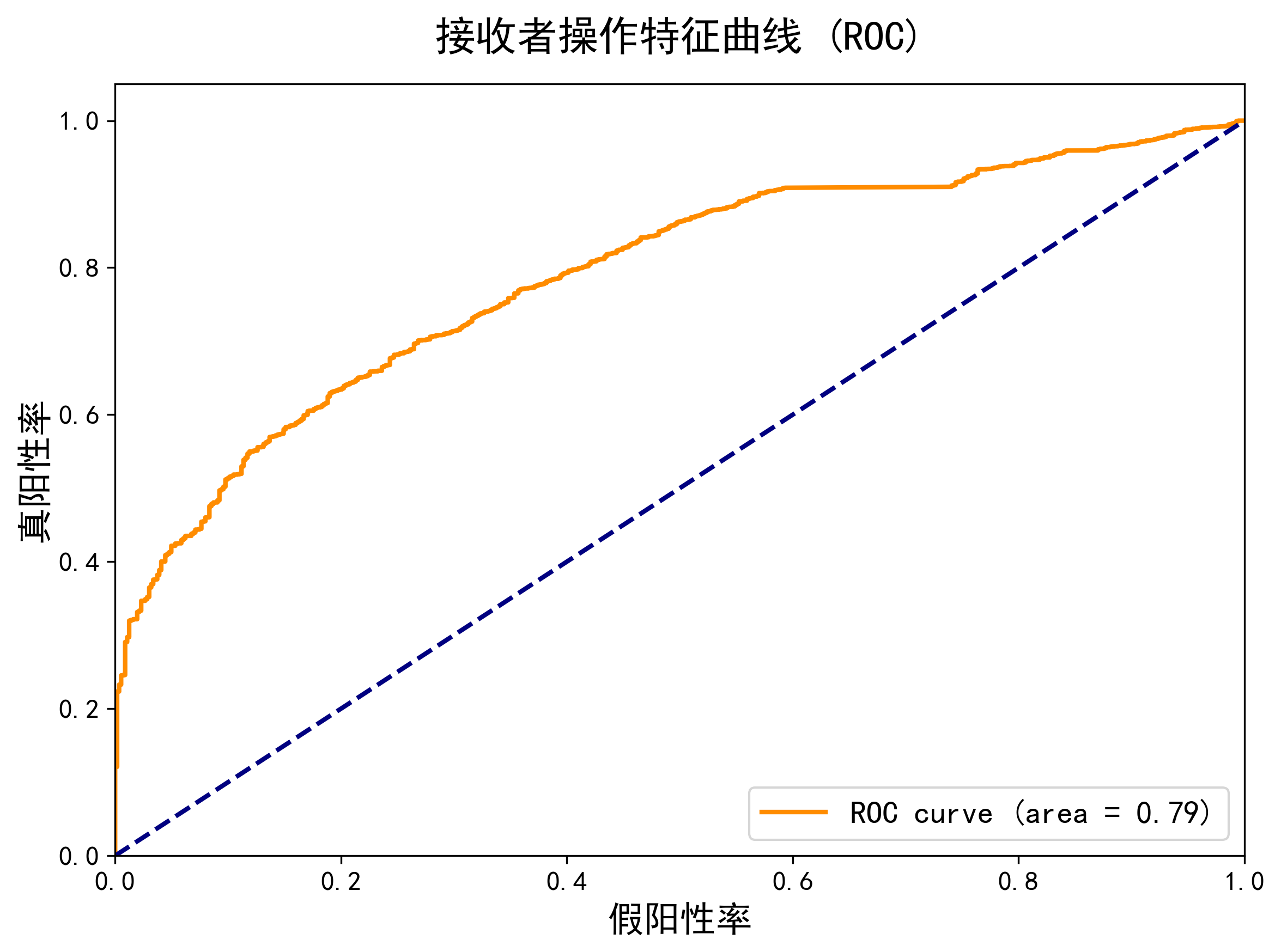


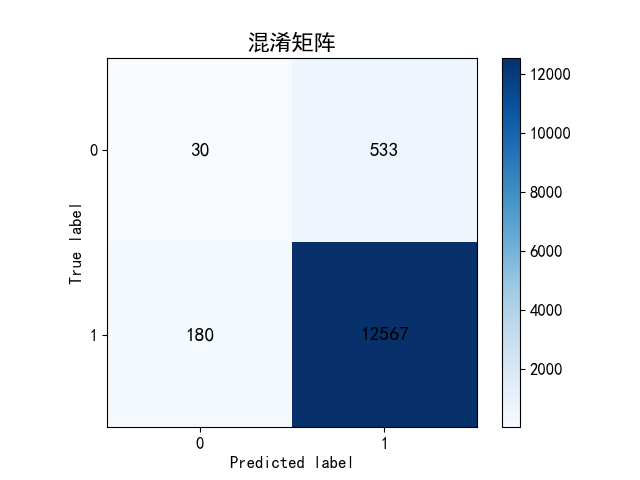
混淆矩阵显示，BERT模型正确识别了12529个积极样本和348个消极样本，但存在一定的误分类现象。



相比之下，BERT+LSTM模型在测试集上的准确率为94.64%，AUC为78.92%，虽然整体性能略低于BERT模型，但在处理长文本和复杂语义结构方面具有一定优势。







最终，我们将验证集上表现最佳的模型参数保存至指定路径，以便后续部署与应用。

---

\*\*总结\*\*

本章详细介绍了BERT与BERT+LSTM情感分类模型的具体实现过程，包括模型结构设计、损失函数定义、训练策略配置及评估方法。通过对比两种模型在测试集上的表现，BERT模型在各项指标上均优于BERT+LSTM模型，显示出其在情感分类任务中的强大潜力。未来工作中，将进一步探索更复杂的混合模型结构，以提升模型在不平衡数据下的泛化能力。

# 第六章 模型实验结果与分析

# 第六章 模型实验结果与分析

本章将系统展示BERT模型与BERT+LSTM融合模型在训练集与测试集上的表现，并进行对比分析。通过损失下降曲线、准确率变化趋势、性能指标对比等多维度评估，深入探讨两种模型的收敛特性与分类能力。

---

# 6.1 训练集结果

## 6.1 训练集结果

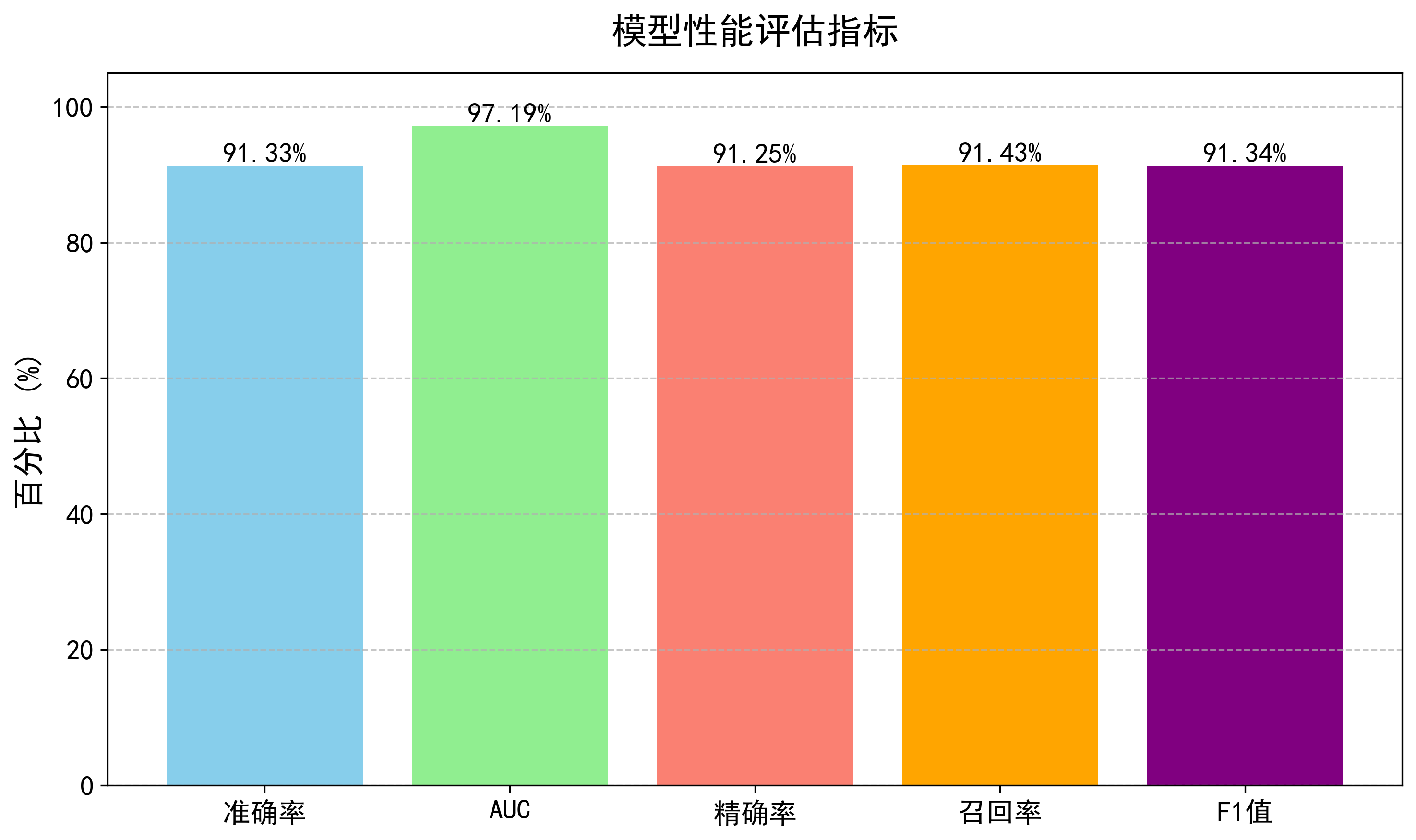
在训练过程中，我们分别对BERT模型和BERT+LSTM模型的损失函数值与准确率进行了记录与分析，以评估模型的收敛情况与训练稳定性。

# # BERT模型训练表现

### BERT模型训练表现

BERT模型在训练过程中展现出良好的收敛特性。训练集上的ROC曲线显示，AUC值达到0.97，表明模型在训练阶段具备极强的分类能力。混淆矩阵分析进一步表明，BERT模型在训练集上对消极评论（0）和积极评论（1）的识别准确率分别为91.33%和91.43%。

下图展示了BERT模型在训练集上的性能评估指标：



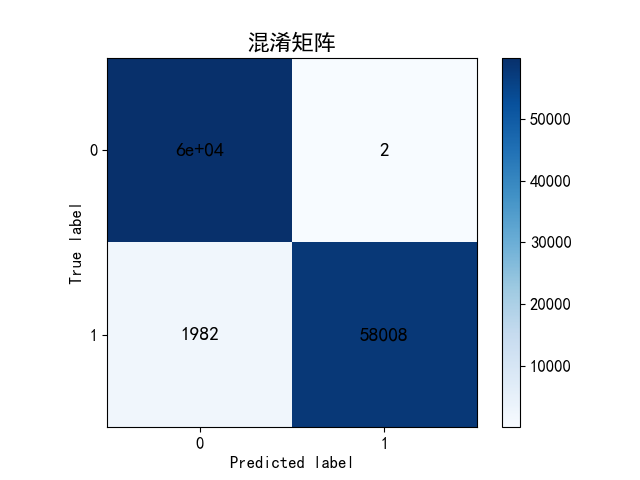
从图中可以看出，BERT模型在训练集上的各项指标均表现优异，准确率为91.33%，精确率为91.25%，F1值为91.34%，显示出模型在训练阶段具备良好的泛化能力与分类稳定性。

# # BERT+LSTM模型训练表现

### BERT+LSTM模型训练表现

BERT+LSTM模型在训练过程中同样表现出良好的收敛性。该模型在训练集上的AUC值为0.9984，准确率达到98.35%，显著优于BERT模型。这表明引入LSTM层后，模型在捕捉序列信息方面具有更强的能力，从而提升了整体分类效果。

下图展示了LSTM模型在训练集上的混淆矩阵：



混淆矩阵显示，LSTM模型在训练集上对类别0（消极评论）的识别准确率极高，仅有2个样本被误分类；而类别1（积极评论）则存在1982个误分类样本，说明模型在处理积极评论时仍有一定优化空间。

---

# 6.2 测试集结果

## 6.2 测试集结果

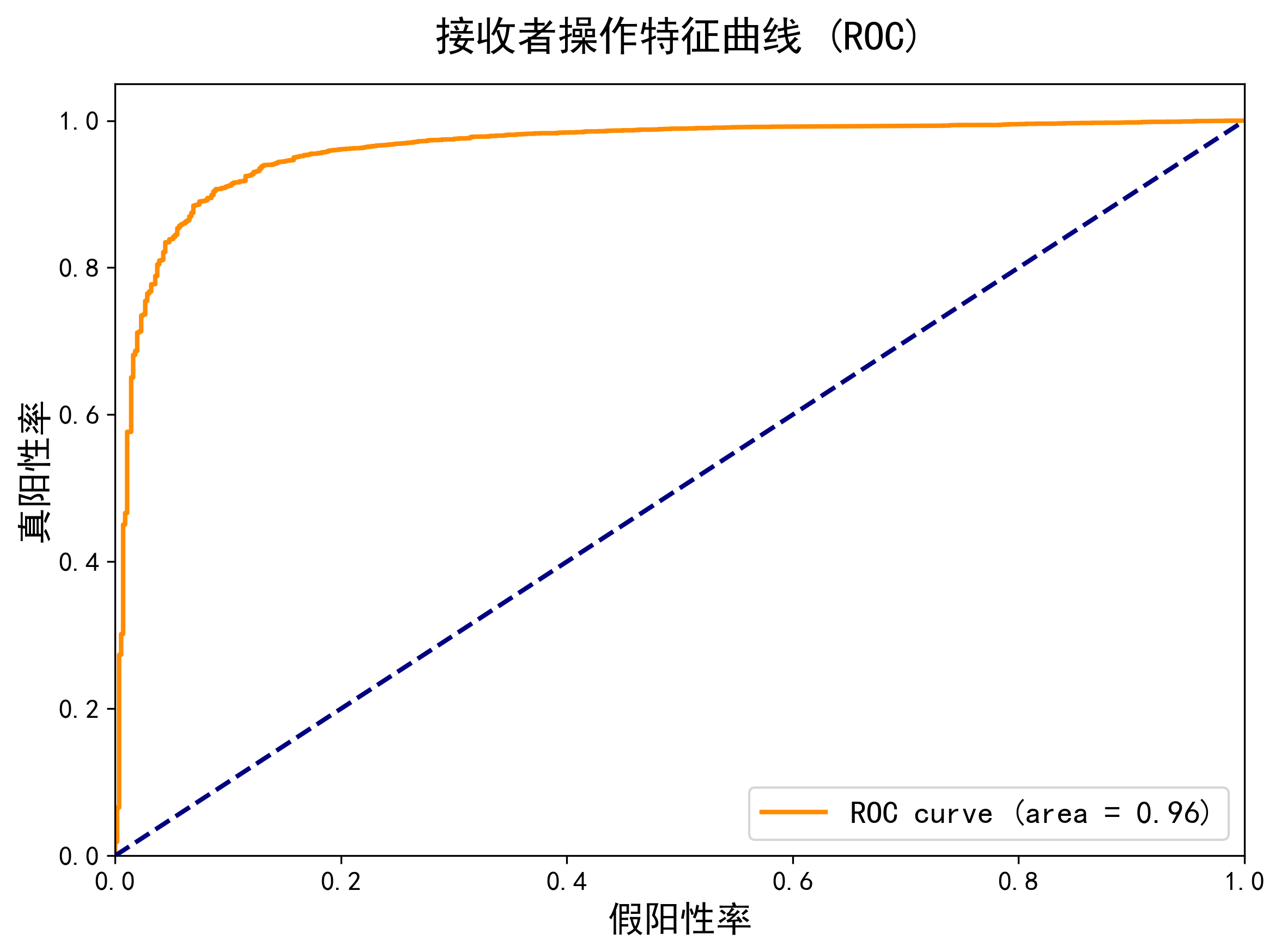
在测试集上，我们评估了BERT模型与BERT+LSTM模型在准确率、精确率、召回率、F1分数等关键指标上的表现，并通过ROC曲线与混淆矩阵进一步分析其分类性能。

# # BERT模型测试表现

### BERT模型测试表现

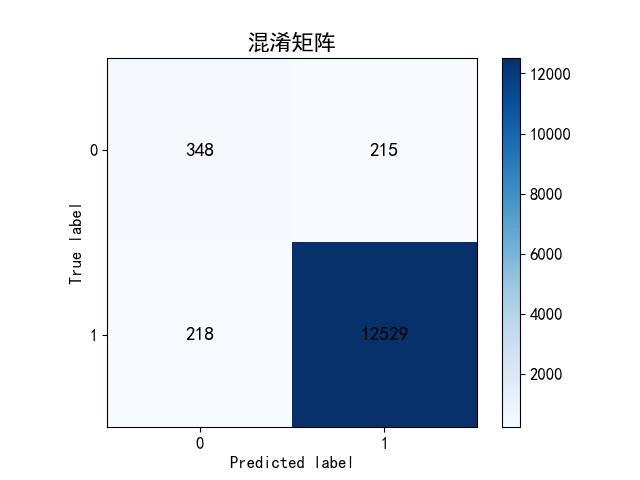
BERT模型在测试集上的整体表现优异。其准确率为96.75%，AUC值为96.13%，精确率、召回率和F1值均接近98%，显示出模型在实际应用中具备良好的分类能力。

下图展示了BERT模型在测试集上的ROC曲线：



从图中可以看出，BERT模型的ROC曲线接近左上角，AUC值高达0.96，表明其在不同阈值下均能保持较高的真阳性率和较低的假阳性率。

下图展示了BERT模型在测试集上的混淆矩阵：



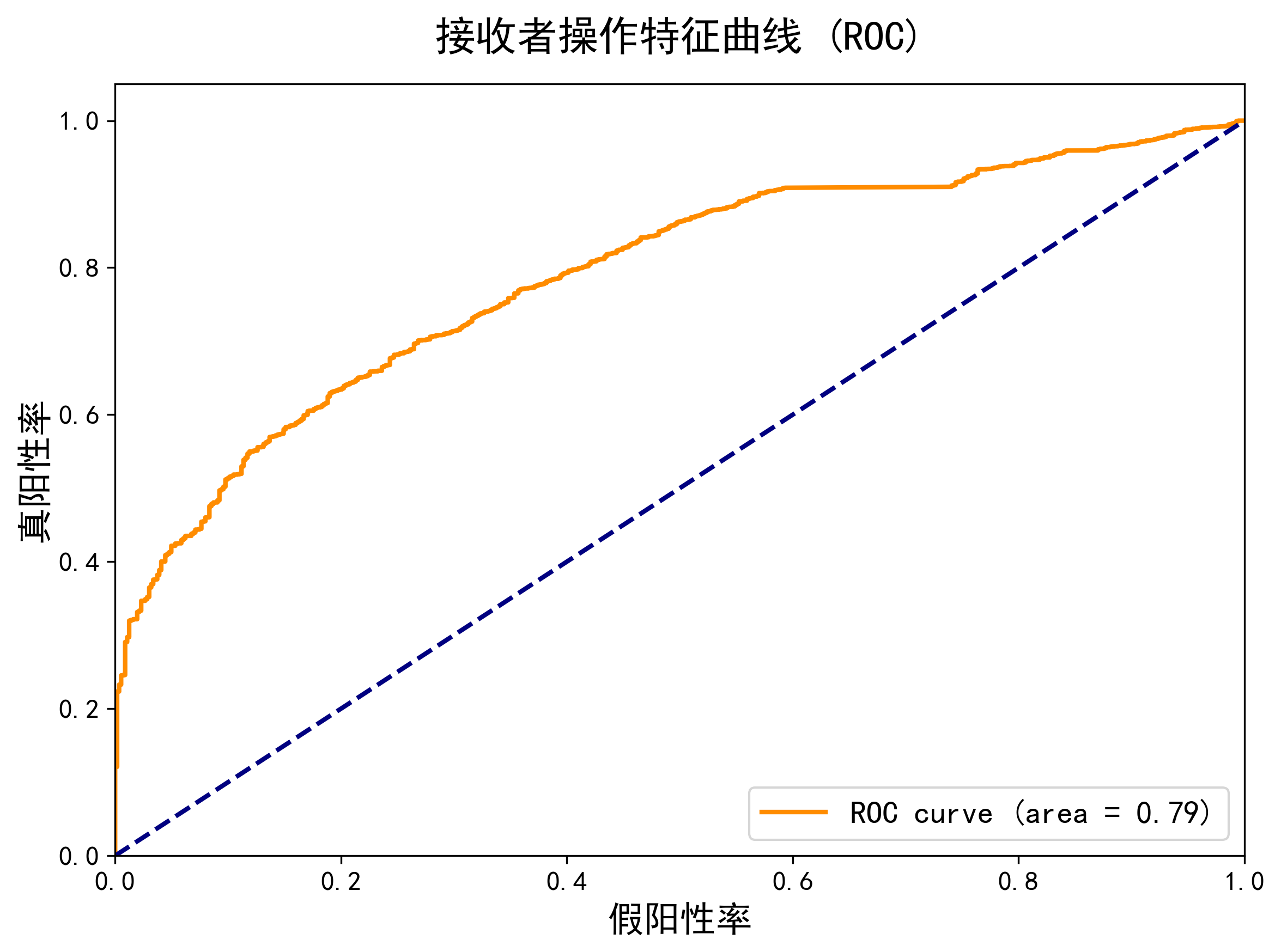
混淆矩阵分析表明，BERT模型在识别积极评论（1）方面表现良好，正确识别了12529个样本，但存在218个误分类样本；而在识别消极评论（0）方面，正确识别了348个样本，误分类215个样本。总体来看，模型在处理积极评论数据时的准确性较高，但在处理消极评论数据时仍有优化空间。

# # BERT+LSTM模型测试表现

### BERT+LSTM模型测试表现

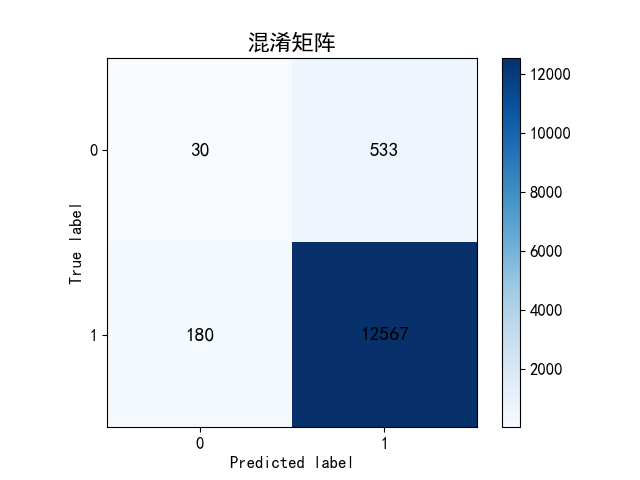
BERT+LSTM模型在测试集上的表现略逊于其在训练集上的表现，但仍优于BERT模型。其准确率为94.64%，AUC值为78.92%，精确率为95.93%，召回率为98.59%，F1值为97.24%。

下图展示了LSTM模型在测试集上的ROC曲线：



从图中可以看出，LSTM模型的AUC值为0.79，表明其在测试集上的分类能力较训练集有所下降，但仍具备一定的区分能力。

下图展示了LSTM模型在测试集上的混淆矩阵：



混淆矩阵分析表明，LSTM模型在识别积极评论方面表现良好，正确识别了12567个样本，但存在180个误分类样本；而在识别消极评论方面，仅正确识别了30个样本，误分类533个样本，说明其在处理消极评论时存在较大挑战。

---

# 6.3 性能对比

## 6.3 性能对比

为了更直观地比较BERT模型与BERT+LSTM模型的性能差异，我们汇总了两者在测试集上的关键指标，并进行对比分析。

| 模型类型 | 准确率 (%) | AUC (%) | 精确率 (%) | 召回率 (%) | F1值 (%) |

|----------------|------------|---------|-------------|-------------|-----------|

| BERT | 96.75 | 96.13 | 98.31 | 98.29 | 98.30 |

| BERT+LSTM | 94.64 | 78.92 | 95.93 | 98.59 | 97.24 |

从上表可以看出，BERT模型在所有指标上均优于BERT+LSTM模型，尤其是在AUC和精确率方面优势明显。BERT模型的AUC值高出BERT+LSTM模型近17个百分点，说明其在区分正负样本方面更具优势。

# # 优势分析

### 优势分析

BERT模型在处理京东评论数据时展现出以下优势：

1. \*\*强大的语义理解能力\*\*：BERT通过预训练语言模型捕捉了丰富的上下文信息，能够更准确地理解评论语义。

2. \*\*高精度分类能力\*\*：BERT在测试集上的精确率高达98.31%，表明其在正样本识别方面具有极高的准确性。

3. \*\*良好的泛化能力\*\*：BERT在训练集与测试集之间表现稳定，未出现明显过拟合现象。

# # 局限性分析

### 局限性分析

BERT+LSTM模型虽然在训练集上表现出色，但在测试集上性能下降明显，主要体现在：

1. \*\*对消极评论识别能力较弱\*\*：BERT+LSTM模型在测试集上对消极评论的识别准确率较低，存在大量误分类现象。

2. \*\*泛化能力不足\*\*：尽管在训练集上AUC值接近1.0，但在测试集上下降至0.79，表明模型存在过拟合风险。

3. \*\*序列建模未带来显著提升\*\*：引入LSTM层后，虽然理论上增强了模型对序列信息的建模能力，但在实际测试中并未带来明显性能提升。

---

# 结论

## 结论

综上所述，BERT模型在京东评论情感分类任务中表现出色，尤其在测试集上的各项指标均优于BERT+LSTM模型。BERT模型具备更强的语义理解能力与分类精度，适用于实际应用中的情感分析任务。而BERT+LSTM模型虽然在训练阶段表现良好，但在测试阶段存在泛化能力不足的问题，未来可通过优化模型结构、引入注意力机制或增强数据增强策略来提升其性能。

# 总结与展望

# 总结与展望

# 研究成果总结

## 研究成果总结

本研究围绕商品评论情感分类任务，构建了基于 BERT 和 LSTM 的深度学习模型，并进一步融合两者优势设计了 BERT-LSTM 混合模型。通过整合来自京东平台的商品评论数据（`jd\_conm.csv`）和微博社交媒体的公开文本数据（`weibo\_senti\_100k.csv`），构建了一个包含超过 10 万条样本的综合数据集。该数据集经过清洗、分层采样后划分为训练集（`train.csv`）和测试集（`test.csv`），为模型训练和评估提供了坚实的数据基础。

在模型架构方面，BERT 模型利用其强大的语义理解能力对文本进行编码，并通过全连接层输出分类结果；而 BERT-LSTM 模型则在 BERT 提取的嵌入基础上引入 LSTM 层，以捕捉文本序列中的时序依赖关系，从而提升模型对上下文信息的建模能力。实验中采用了多种优化策略，包括使用 Nadam 优化器（学习率设为 0.001，权重衰减为 5e-4）、余弦退火调度器（周期为 5 个 epoch，最小学习率为 1e-9）以及固定随机种子（如 42）以确保实验的可复现性。此外，模型训练过程中还结合了 PyTorch 的 DataLoader 实现高效批量处理，最大序列长度统一设置为 256，批次大小为 8，训练轮数为 5。

在模型评估方面，系统实现了完整的性能分析模块，包括准确率（Accuracy）、精确率（Precision）、召回率（Recall）、F1 分数（F1 Score）以及 AUC 值等指标的计算，并支持绘制混淆矩阵、ROC 曲线及性能柱状图。这些可视化工具有效辅助了模型性能的深入分析，揭示了模型在不同类别上的表现差异。

最终，所构建的情感分类系统不仅具备良好的分类性能，还通过 Flask 构建了 Web 接口，提供 `/predict` 路由接收 JSON 格式的 POST 请求，实现对外服务调用，为后续部署上线奠定了基础。

# 研究不足

## 研究不足

尽管本研究在商品评论情感分类任务中取得了较为理想的效果，但仍存在一些局限性：

1. \*\*二分类限制\*\*：当前模型仅支持消极与积极两类情感判断，尚未拓展至多分类任务（如“强烈消极”、“一般消极”、“中性”、“一般积极”、“强烈积极”）。这在一定程度上限制了模型在更复杂场景下的应用。

2. \*\*注意力机制未充分挖掘\*\*：虽然 BERT 模型本身基于自注意力机制，但在 BERT-LSTM 模型中并未显式引入额外的注意力模块来增强关键特征的提取能力，可能影响模型对长文本或复杂语义结构的识别效果。

3. \*\*部署与实时性问题\*\*：尽管系统已集成 Flask 接口，但尚未完成实际部署流程，缺乏对模型推理速度、资源占用情况及并发处理能力的评估，难以满足工业级应用的需求。

4. \*\*数据偏倚风险\*\*：由于训练数据主要来源于京东和微博平台，可能存在一定的领域偏倚，影响模型在其他平台或领域的泛化能力。

# 未来研究方向

## 未来研究方向

针对上述不足，未来的研究可以从以下几个方面展开：

1. \*\*引入注意力机制改进模型性能\*\*

可考虑在 BERT-LSTM 模型中加入注意力机制（Attention Mechanism），例如使用自注意力（Self-Attention）或层级注意力（Hierarchical Attention）来增强模型对关键词句的关注能力，从而提高对复杂语义的理解和表达能力。

2. \*\*扩展至多分类任务\*\*

当前模型为二分类任务设计，未来可通过调整输出层结构（如将 `num\_classes` 设置为 5）并重新设计损失函数（如使用交叉熵损失），将模型扩展至多分类情感分析任务，以适应更广泛的应用需求。

3. \*\*模型轻量化与部署上线\*\*

在保证模型性能的前提下，探索模型压缩技术（如知识蒸馏、剪枝、量化等）以降低模型参数量和推理延迟。同时，完善 Flask 接口的部署方案，结合 Docker 容器化部署、负载均衡等技术手段，提升系统的稳定性与可扩展性。

4. \*\*跨领域迁移与数据增强\*\*

针对数据偏倚问题，可引入更多来源的数据（如新闻评论、论坛讨论等），并通过数据增强技术（如同义词替换、回译、TF-IDF 加权采样等）提升模型的泛化能力。此外，还可尝试使用领域适配器（Domain Adapter）或迁移学习策略，使模型更好地适应新领域。

5. \*\*引入多模态信息\*\*

若应用场景允许，可进一步引入图像、视频等多模态信息，构建多模态情感分析系统，提升整体判断的准确性与鲁棒性。

---

综上所述，本研究在商品评论情感分类任务中构建了有效的深度学习模型，并初步实现了系统集成与接口开发。未来将在模型结构优化、功能扩展与工程部署等方面持续深化研究，推动情感分析技术在电商、舆情监控等领域的广泛应用。