人工智能基础 (A) 大作业 实验报告

项目名称： 基于BERT的新闻标题情感分析系统

日期： 2025 年 6 月 9 日

代码详见仓库：BERT-News-Analyzer: 基于BERT的新闻标题情感分析 (https://github.com/Fraserrr/BERT-News-Analyzer)

摘要

本项目旨在设计并实现一个基于深度学习的新闻标题情感分析系统。系统利用了当前自然语言处理（NLP）领域先进的 BERT（Bidirectional Encoder Representations from Transformers）系列模型，通过微调（Fine-Tuning）技术使其适应新闻标题的情感分类任务（正面、中性、负面）。系统前端采用 Gradio 框架构建，提供了一个交互式的 Web 用户界面，用户不仅可以对输入的单个新闻标题进行实时的情感预测，还可以自主选择不同的 BERT 基座模型，调整关键超参数，在线训练和评估模型。本项目完整覆盖了从数据处理、模型训练、性能评估到最终部署应用的全过程，实现了技术研究与工程实践的结合。

关键词： 情感分析；BERT；深度学习；模型微调；Gradio；自然语言处理

一、 引言与背景

自然语言处理（NLP）是人工智能领域的重要分支，其核心目标是让计算机能够理解、解释和生成人类语言。近年来，以 Transformer 模型为代表的深度学习架构彻底改变了NLP领域。在此基础上诞生的BERT模型，通过其创新的预训练-微调范式，在多项NLP任务中取得了突破性进展。

新闻标题作为新闻内容的精炼概括，其情感倾向直接影响着信息的传播和用户的感知。快速准确地分析新闻标题的情感，对于舆情监控、推荐系统、公众情绪洞察等领域具有重要的应用价值。

本项目以此为背景，旨在构建一个功能完善、操作便捷的新闻标题情感分析系统。系统不仅要实现高精度的情感分类功能，还要为使用者提供模型训练的灵活性，使其能够直观地体验和对比不同BERT模型在特定任务上的表现。

1.1 系统目标

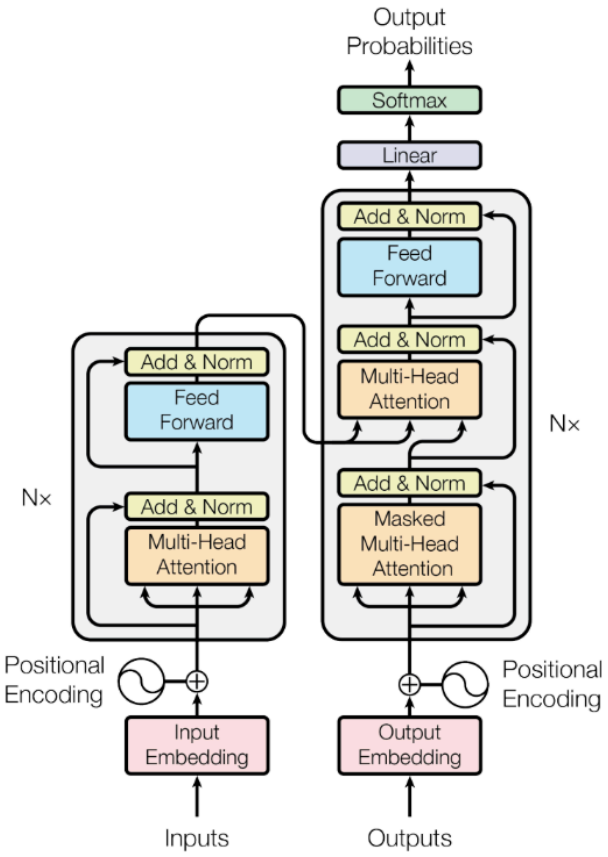
1. **实现自动化情感分析**：对输入的新闻标题文本，系统能够自动判断其情感倾向（正面、中性、负面），并给出各类别的置信度。
2. **支持多种BERT模型**：系统集成多种主流的中文预训练BERT模型，用户可以根据需求选择不同的基座模型进行微调和分析。
3. **提供模型在线训练与评估**：用户可以通过UI界面调整学习率等核心超参数，启动模型训练流程，并在训练结束后查看包括损失曲线、准确率、F1分数、ROC曲线在内的详细评估报告。
4. **构建用户友好的交互界面**：利用 Gradio 搭建 Web UI，将复杂的技术流程封装在简洁的操作背后，提供直观的结果可视化。

二、 相关技术原理

2.1 从Transformer到BERT

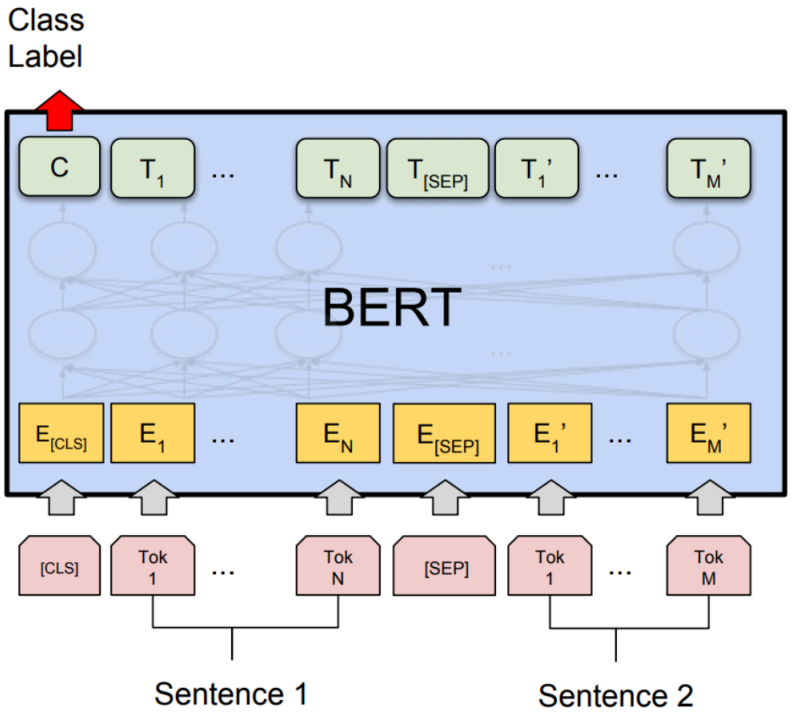
本项目核心技术基于BERT模型，其理论基础源于Transformer架构。

- **Transformer**：它是一种完全基于自注意力（Self-Attention）机制的深度神经网络架构。相较于传统的循环神经网络（RNN）或卷积神经网络（CNN），Transformer能够更好地捕捉文本中的长距离依赖关系，并支持并行计算，极大地提高了模型训练的效率和性能。



2.2 BERT模型核心思想

BERT（Bidirectional Encoder Representations from Transformers）的字面意思是“来自变换器的双向编码器表示”。它的核心创新在于解决了传统语言模型单向性的问题，能够真正理解一个词在句子中的确切含义，因为它同时考虑了该词左右两侧的上下文信息。



2.3 预训练-微调范式

BERT的强大威力来源于其“两阶段”工作模式：预训练（Pre-training）和微调（Fine-tuning）。

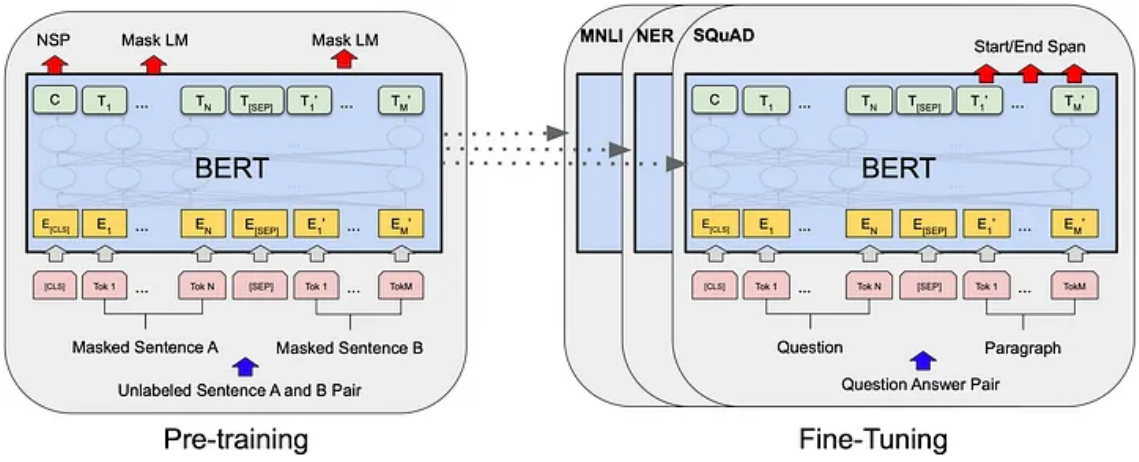
1. **预训练 (Pre-training):**

   BERT首先在海量的无标签文本数据（如维基百科、书籍语料）上进行训练，学习通用的语言规律。这个阶段主要完成两个任务：

   - 掩码语言模型 (Masked Language Model, MLM)：随机遮盖句子中的部分单词，然后让模型去预测这些被遮盖的单词。这迫使模型学习单词间的双向依赖关系和深层次的语义表示。
   - 下一句预测 (Next Sentence Prediction, NSP)：给定两个句子A和B，让模型判断B是否是A的下一句。这个任务让模型学习句子间的逻辑关系。

2. **微调 (Fine-tuning):**

   预训练好的BERT模型已经具备了强大的语言理解能力。针对特定的下游任务（如情感分类），我们不再需要从零开始训练模型。只需在预训练模型的基础上，加入一个简单的分类层，然后使用我们自己的、规模相对较小的有标签数据集（如本项目中的新闻标题情感数据集）对整个模型进行“微调”。模型参数会根据新任务进行少量调整，从而快速适应并获得优异的性能。

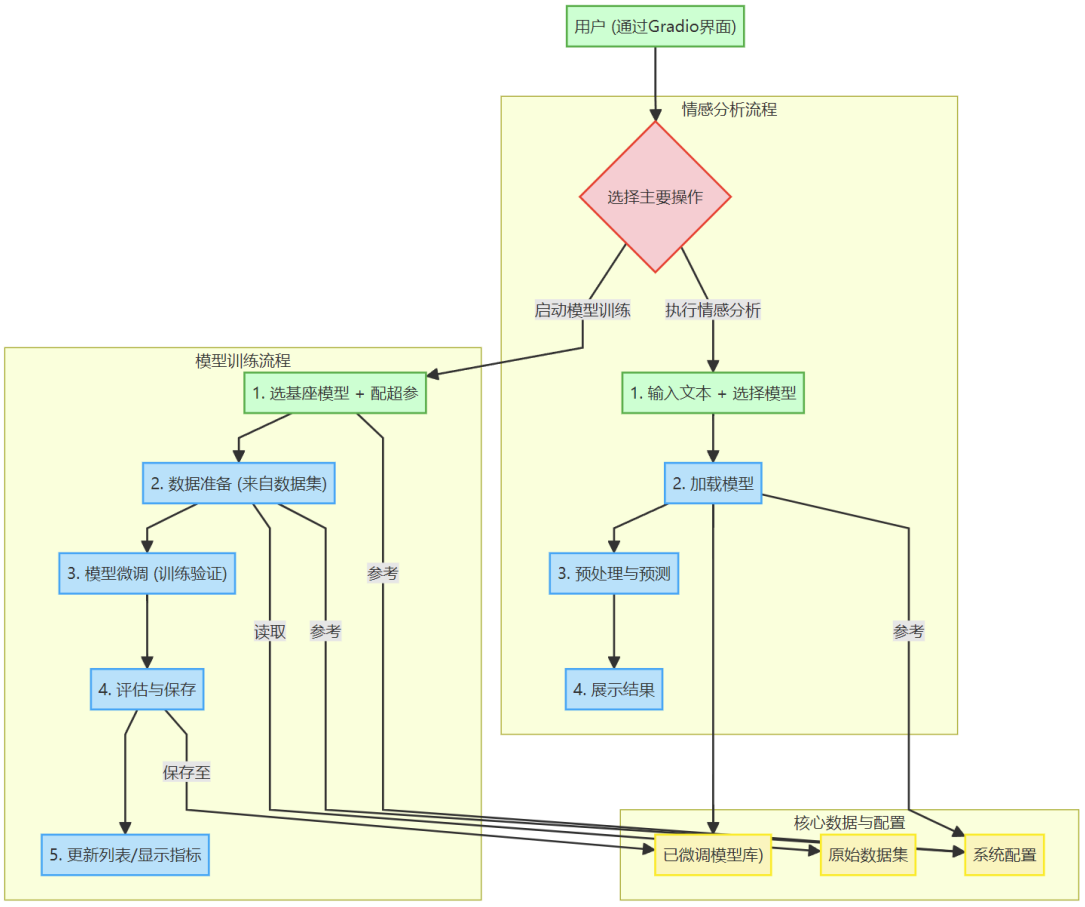


三、 系统设计与实现

本系统遵循模块化的设计思想，将数据处理、模型训练、UI交互等功能解耦，代码结构清晰，易于维护和扩展。

3.1 系统整体流程

系统的工作流程分为两大核心部分：模型训练流程和情感分析流程。两者共享底层的模型库、数据集和配置文件。用户通过 Gradio 界面发起操作，后端逻辑根据用户的选择调用相应的功能模块。



3.2 项目代码结构

项目的代码文件组织如下：

```
.
├── app.py                # Gradio应用主文件，负责UI构建与交互逻辑
├── model_trainer.py      # 模型训练与评估模块
├── common_utils.py       # 通用工具模块，如数据加载、Dataset定义等
├── config.py             # 配置文件，存储常量和路径
├── dataset/
|   └── AIA-project2-dataset.csv # 项目数据集
├── user_trained_models/ #  （运行时自动创建）保存微调后的模型
├── user_training_logs/  #  （运行时自动创建）保存训练日志
└── README.md             # 项目说明文档
```

3.3 数据处理模块 (`common_utils.py`)

数据处理是模型训练的第一步，其主要流程如下：

1. **加载数据**：使用`pandas`库从`dataset/AIA-project2-dataset.csv`文件中读取新闻标题（`text`列）和情感标签（`label`列）。
2. **标签映射**：为了便于模型计算，将文本标签（如 "positive", "negative", "neutral"）映射为数字ID（如 2, 0, 1）。这一映射关系定义在`LABEL_MAP`和`ID_TO_LABEL`中。
3. **数据清洗**：移除标签或文本缺失的行，确保数据质量。
4. **创建 PyTorch Dataset**：定义一个`SentimentDataset`类，继承自`torch.utils.data.Dataset`。在该类中，使用BERT模型对应的`Tokenizer`对文本进行分词、编码、截断和填充，将其转换为模型所需的`input_ids`和`attention_mask`张量。
5. **创建 DataLoader**：使用`DataLoader`将`SentimentDataset`封装成可迭代的数据加载器，实现批处理（Batching）和数据打乱等功能。

3.4 模型训练与评估模块 (`model_trainer.py`)

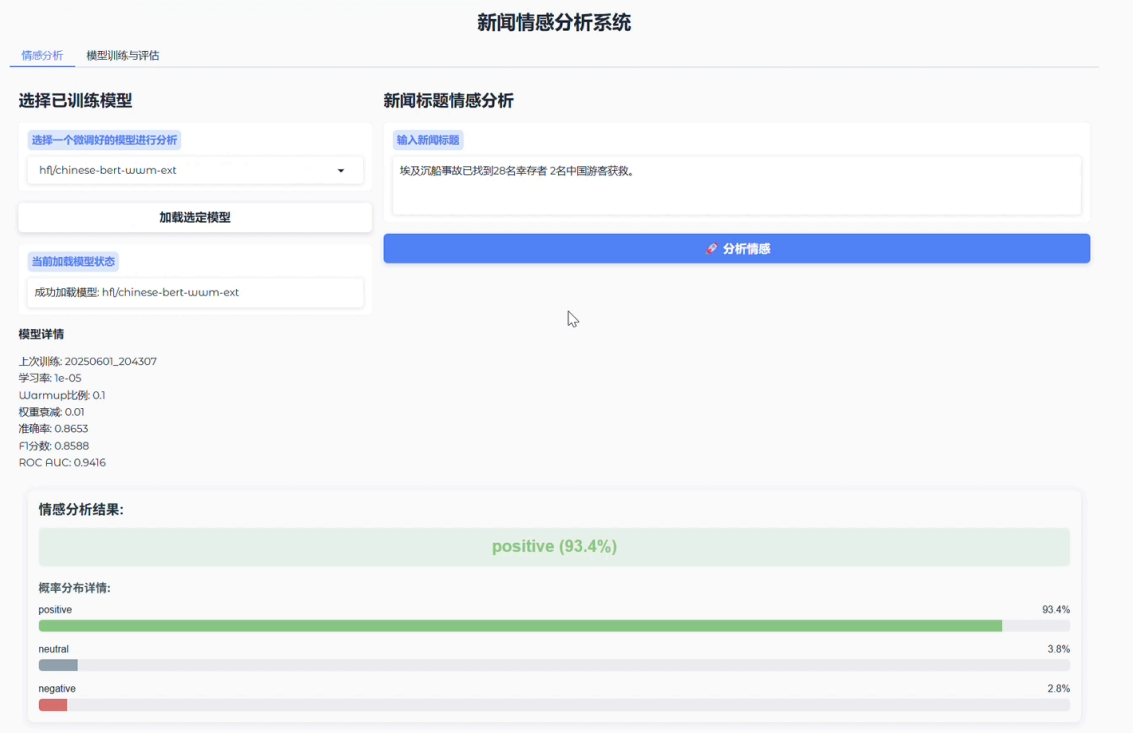
这是系统的核心计算模块，负责模型的微调和性能评估。

1. **模型选择**：系统支持从 `config.py` 中定义的 `MODELS_TO_TEST` 列表中选择基座模型，包括：
   - `bert-base-chinese`
   - `hfl/chinese-bert-wwm-ext`
   - `hfl/chinese-roberta-wwm-ext`
   - `hfl/chinese-macbert-base`
2. **数据划分**：使用`sklearn.model_selection.train_test_split`将数据集划分为训练集、验证集和测试集，以进行模型训练和公正的性能评估。
3. **训练循环**：
   - 加载预训练的BERT模型和`Tokenizer`。
   - 定义优化器（`AdamW`）和学习率调度器（`get_scheduler`）。
   - 在设定的`Epoch`数内进行迭代训练。在每个`batch`中，将数据送入模型进行前向传播，计算损失（`loss`），然后进行反向传播和参数更新。
   - 在每个`Epoch`结束后，在验证集上进行评估，并保存验证集上性能最好的模型。
4. **性能评估**：
   - 训练结束后，在测试集上评估最终模型的性能。
   - 计算并记录多项指标，包括准确率（Accuracy）、精确率（Precision）、召回率（Recall）、F1分数（F1-Score）和 ROC AUC。
5. **结果保存**：
   - 将微调后的模型权重和`Tokenizer`配置文件保存到`user_trained_models`目录。
   - 将本次训练的超参数、评估指标、损失历史等详细信息保存为`training_info.json`文件，便于溯源和分析。

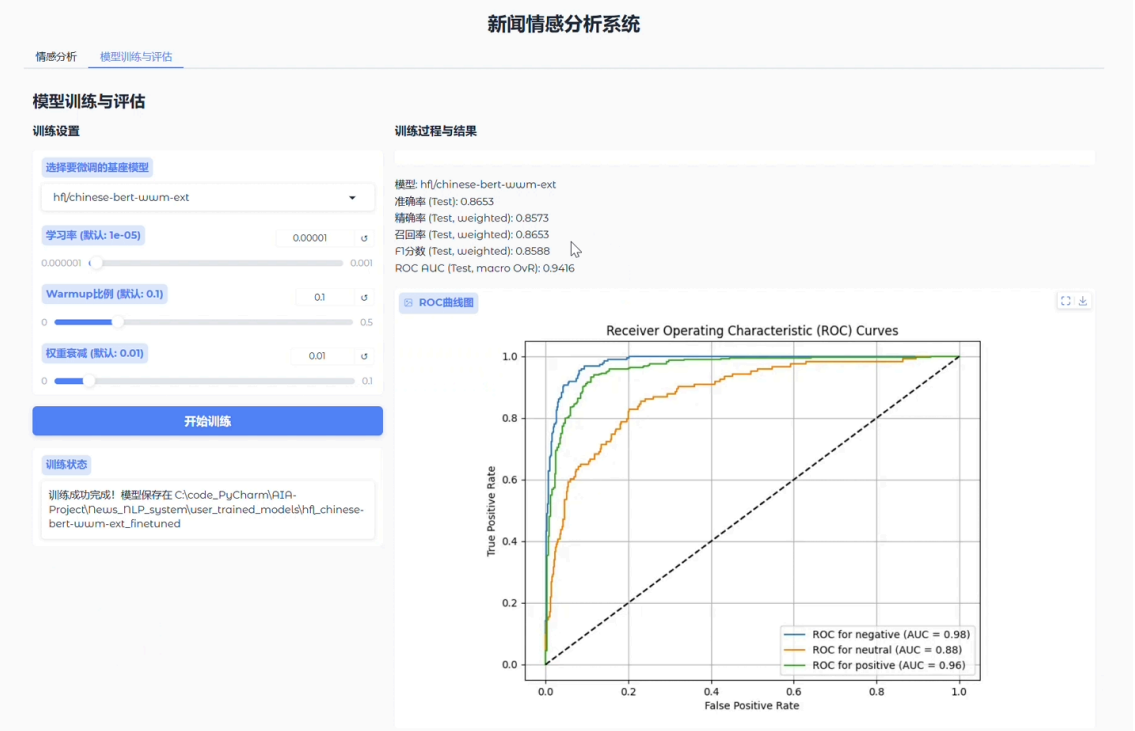
3.5 用户界面模块 (`app.py`)

本系统采用 Gradio 构建 Web 界面，为用户提供直观的操作体验。界面主要包含两个选项卡。

1. **情感分析选项卡**：
    - 模型选择：一个下拉菜单，列出所有`user_trained_models`中已训练好的模型。
    - 模型加载：点击按钮加载选定的模型，并显示其状态和训练详情。
    - 文本输入：一个文本框，供用户输入待分析的新闻标题。
    - 结果展示：分析结果以HTML格式动态生成，通过不同颜色和百分比条清晰地展示预测的情感类别及其置信度。



2. **模型训练与评估选项卡**：
    - 训练设置：提供下拉菜单选择基座模型，以及滑块来调整学习率、Warmup比例和权重衰减等超参数。
    - 启动训练：点击按钮即可根据当前设置启动后台训练任务，并实时显示训练状态和进度条。
    - 结果可视化：训练完成后，界面会展示最终的评估指标、训练/验证损失曲线图，以及ROC曲线图。



四、 系统功能与特色

4.1 实时情感分析与可视化

系统最核心的功能是提供即时的情感分析服务。用户输入文本后，点击分析按钮，即可在数秒内获得直观的分析结果。结果的可视化设计是本系统的一大特色：

- 主情感突出显示：最可能的情感类别会被放大并用醒目的颜色（如绿色代表正面，红色代表负面）高亮展示。
- 概率分布条：所有情感类别的概率都通过一个条形图展示，用户可以清晰地看到模型对不同情感的置信度对比。

4.2 灵活的模型训练与对比

本系统不仅是一个应用工具，也是一个实验平台。

- 模型选择的灵活性：用户可以一键切换不同的中文BERT模型进行训练，直观对比它们在同一数据集上的性能差异。
- 超参数可调：开放了学习率等关键超参数的调节接口，让使用者可以探索不同参数组合对模型性能的影响。
- 全面的评估报告：训练结束后自动生成的损失曲线和ROC曲线等图表，为模型评估提供了充分的数据支持，帮助用户深入理解模型训练过程和结果。

4.3 便捷的部署与使用

得益于 Gradio 框架，整个系统的启动和访问较为便捷。

1. 环境配置：根据 `requirements.txt` 安装所有Python依赖库。
2. 数据集准备：确保 `AIA-project2-dataset.csv` 文件位于 `dataset` 目录下。
3. 一键启动：在终端运行 `python app.py` 命令，即可启动Web服务。
4. 浏览器访问：通过本地或局域网IP地址即可访问系统界面。

五、 实验与结果分析

为验证模型在新闻标题情感分析任务上的有效性，并寻找最优的模型与参数组合，我们进行了一系列对比实验。实验使用课程提供的数据集 `AIA-project2-dataset.csv` 。

5.1 实验模型选择

我们选取了当前中文自然语言处理领域中几个主流且有代表性的预训练模型作为微调的基座，包括：

- `bert-base-chinese`：Google发布的原版中文BERT模型，作为性能基准。
- `hfl/chinese-bert-wwm-ext`：哈工大讯飞联合实验室发布的模型，采用了全词掩码（WWM）技术和更大的语料库，在许多任务上表现出色。
- `hfl/chinese-roberta-wwm-ext`：在BERT基础上改进的RoBERTa模型，训练策略更优。
- `hfl/chinese-macbert-base`：同样来自哈工大讯飞，通过改进掩码策略（MLM as Correction）提升性能。
- `nghuyong/ernie-3.0-base-zh`：百度提出的ERNIE模型，通过融入知识图谱信息增强了语言表示能力。

通过对这些不同架构和训练策略的模型进行比较，我们可以更全面地评估哪种模型最适合本项目的数据和任务。

5.2 超参数调优

在确定最终的对比实验参数前，我们以表现突出的 `hfl/chinese-bert-wwm-ext` 模型为例，进行了一系列超参数的探索实验。目的是找到一个能够让模型快速收敛且性能优异的通用参数组合。下表记录了部分关键的探索过程：

学习率 (LR)	训练轮数 (Epochs)	批大小 (Batch Size)	权重衰减 (WD)	测试集准确率
`3e-05`	5	16	0.01	83.98%
`1e-05`	5	16	0.01	84.91%
`2e-05`	3	16	0.01	85.00%
`2e-05`	3	32	0.01	85.74%
`1e-05`	**3**	**16**	**0.01**	**86.53%**

调优分析与结论：

1. **学习率**：实验表明，`3e-05` 的学习率相对较高，模型性能不如 `1e-05` 或 `2e-05`。较低的学习率能让模型在微调阶段更稳定地收敛到最优解。
2. **训练轮数**：训练 5 个轮次时，在第 4 或第 5 轮的验证准确率有时不再提升，甚至验证损失上升这表明模型有出现过拟合的风险。而训练 3 个轮次在性能和效率之间取得了较好的平衡。
3. **批大小**：增大批大小到 32 能提升准确率至 85.74%，但考虑到显存消耗和训练稳定性，16 的批大小在多数实验中已表现出优异且稳健的性能。
4. **最终选择**：综合考虑，我们确定了 学习率为 `1e-05`，训练轮数为 `3`，批大小为 `16`，权重衰减为 `0.01` 作为后续所有模型进行横向对比的“最优超参数组合”。在这个组合下，`hfl/chinese-bert-wwm-ext` 取得了 86.53% 的测试集准确率。

5.3 不同模型性能横向对比

我们使用上一节确定的最优超参数组合，对所有选定的基座模型进行了微调，并在相同的测试集上评估其性能。结果如下表所示：

基座模型	测试集准确率	最佳验证集准确率	备注
hfl/chinese-bert-wwm-ext	**86.53%**	84.03%	**综合表现最优**
nghuyong/ernie-3.0-base-zh	86.39%	83.89%	性能非常接近，表现优异
bert-base-chinese	86.11%	83.19%	表现稳健，可作为基线
hfl/chinese-macbert-base	85.69%	84.03%	性能良好
hfl/chinese-roberta-wwm-ext	85.42%	85.00%	验证集表现好，但测试集稍逊

结果分析：

- `hfl/chinese-bert-wwm-ext` 在本次实验中表现最佳。这很可能得益于其在预训练阶段采用的“全词掩码（Whole Word Masking）”技术和更大规模的中文语料库，使其对中文的理解更为深刻，在微调后能更好地适应本任务。
- `nghuyong/ernie-3.0-base-zh` 和 `bert-base-chinese` 也取得了极具竞争力的结果，证明了它们作为通用中文预训练模型的强大能力。
- `hfl/chinese-roberta-wwm-ext` 和 `hfl/chinese-macbert-base` 虽然在其他任务上可能具备优势，但在本次实验的特定设置下，准确率略低于前三者。

5.4 定性分析与泛化能力测试

基于表现最优的 `hfl/chinese-bert-wwm-ext` 模型，我们从数据集外选取了几个例子进行泛化能力测试：

- **输入**：“埃及沉船事故已找到28名幸存者 2名中国游客获救。” -> **预测结果：正面 (Positive)**
  - 分析：模型准确捕捉到了“幸存者”、“获救”等积极词汇，给出了正面判断，符合人类认知。
- **输入**：“美国国债突破36万亿美元 债务螺旋带来严峻考验！” -> **预测结果：负面 (Negative)**
  - 分析：模型识别了“突破”、“债务螺旋”、“严峻考验”等词语带来的负面情绪，判断准确。
- **输入**：“罗马天主教教皇方济各去世，享年88岁” -> **预测结果：中性 (Neutral)**
  - 分析：模型将“去世”这一客观事实陈述判断为中性，而非负面，表现出较高的语义理解能力。

实验结果表明，经过微调的BERT模型能够准确地捕捉新闻标题中的情感信息，且预测能力不局限于训练数据集，具备良好的泛化能力，符合项目预期目标。

六、 总结与展望

本项目成功设计并实现了一个集情感分析、模型训练与评估功能于一体的 Web 系统。通过结合强大的 BERT 模型和友好的 Gradio 界面，我们打造了一个兼具实用性和可扩展性的 NLP 应用。

总结：

- 技术层面：验证了BERT模型在中文新闻标题情感分类任务上的有效性，并实现了完整的"预训练-微调"应用流程。
- 工程层面：构建了一个模块化、易于部署和使用的系统，实现了从后端算法到前端交互的闭环。

展望：

- 模型优化：未来可以尝试更先进或更轻量级的模型（如ELECTRA, ALBERT），并引入更复杂的超参数优化策略。
- 功能扩展：系统可以扩展到其他文本分析任务，如实体识别、文本摘要、主题分类等。
- 数据增强：引入更多样化的数据集或采用数据增强技术，进一步提升模型的泛化能力。

七、附录

7.1 开放源代码

本项目工程代码已开源在 Github 平台，仓库链接：

BERT-News-Analyzer: 基于BERT的新闻标题情感分析 (https://github.com/Fraserrr/BERT-News-Analyzer)

7.2 小组成员贡献表

姓名	主要工作	贡献占比