

学 号:	1221004042
------	------------



# 课程报告

题 目	基于python和spark的文本分类
学 院	理学院
专 业	数据科学与大数据技术
班 级	大数据222
姓 名	张徐烨

2025 年\_6\_ 月\_20\_ 日

## 摘 要

本文针对自然语言处理中的句子级分类任务，采用 BERT 模型进行深层语义特征提取，并结合传统机器学习中的逻辑回归（Logistic Regression, LR）分类器，完成对 GLUE 数据集中的 CoLA 和 SST-2 子任务的建模与评估。通过在 PySpark 框架中构建分布式特征处理与模型训练流程，有效提升了计算效率。在特征工程、参数调优与结构设计方面进行多次优化，使得 CoLA 的 Matthews 相关系数（MCC）从初始的 0.22 提升至 0.4067，SST-2 的准确率达到 86.01%。本文详细介绍数据预处理、特征提取、模型结构、训练优化及评估指标，并分析该模型在实际落地场景下的潜力与扩展。

**关键字：**BERT；GLUE；Spark；逻辑回归；Matthews

## 目 录

1 引言 .....	4
2 国内外研究现状.....	4
3 特征工程.....	4
3.1 原始文本表示 .....	4
3.2 BERT 表达特征 .....	4
3.3 特征处理方式 .....	4
4 模型结构分析.....	5
4.1 总统结构框架 .....	5
4.2 模型组件说明 .....	6
4.3 其他优化细节 .....	6
5 数据处理与实验设置.....	7
5.1 数据预处理 .....	7
5.2 内存优化 .....	7
6 实验结果与分析.....	7
6.1 CoLA 任务结果 .....	7
6.2 SST-2 任务结果 .....	8
6.3 效果分析 .....	8
7 结论与展望.....	8
参考文献.....	9

# 1 引言

随着深度学习的发展，基于 Transformer 的语言模型已成为 NLP 的核心工具。BERT 通过双向 Transformer 编码器，在多个下游任务中表现卓越。然而直接使用 BERT 进行分类面临显存开销大、微调成本高等问题，本文采用轻量级策略，即冻结 BERT 权重，仅使用其提取的句子特征训练传统的 LR 分类器，在保证效果的同时降低训练成本。本研究任务选取了 GLUE 基准中的 CoLA（语法接受性判断）与 SST-2（情感分析）两个子任务，在 PySpark 分布式环境中结合 BERT 表达能力与逻辑回归的判别性，构建高效的轻量级文本分类系统。

## 2 国内外研究现状

国外如 Google、Facebook 等机构常采用 BERT + 分类器作为通用微调基线，但多数使用完整 fine-tuning；国内高校与企业更重视部署效率与轻量模型，常采用冻结 BERT + 简单分类器策略；本文采用方案兼具迁移能力、部署友好、训练可控三者平衡，适合工业实际应用。

## 3 特征工程

### 3.1 原始文本表示

- 每个样本为一句英文句子（例如：“Fred watered the plants flat.”）。
- CoLA 数据集每条样本标注是否为语法正确（0/1），SST-2 则标注情感极性（0=负面，1=正面）。

### 3.2 BERT 表达特征

- 使用 bert-base-uncased 作为预训练模型，具体配置：
  - 12 层 Transformer 编码器
  - 12 个注意力头
  - 隐藏层维度 768
  - 词表大小 30522
  - 最大序列长度 512
- 特征提取过程：
  - 批处理大小为 32，使用 torch.no\_grad() 节省内存
  - 使用 BERT tokenizer 进行分词，支持自动填充（padding）和截断（truncation）
  - 输入包含 input\_ids 和 attention\_mask 两个张量
  - 提取最后一层 [CLS] token 的隐藏状态作为句子表示
- 模型参数保持冻结状态（model.eval()），仅作为编码器；
- 除了 BERT 的 [CLS] 向量，还添加了“句子长度”作为一个附加维度，最终特征维度为 769: 768（来自 BERT）+ 1（长度）。

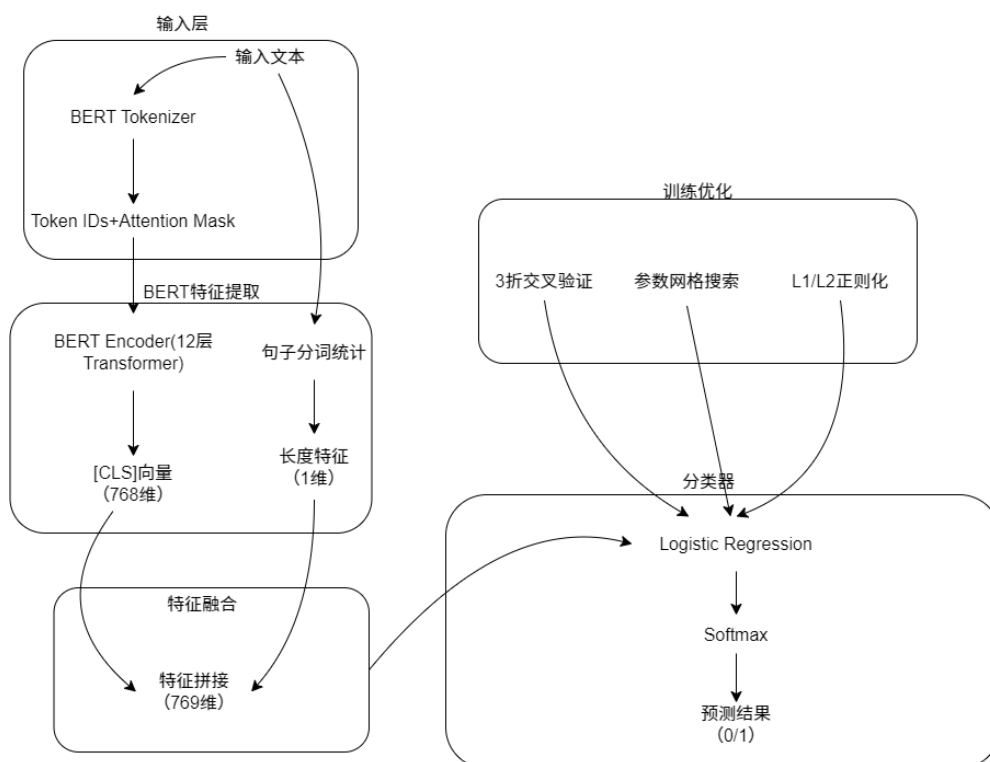
### 3.3 特征处理方式

- 特征提取后使用 Pandas 组装为 DataFrame，并与标签拼接；

- 转换为 Spark DataFrame，并通过 VectorAssembler 转化为向量字段 features；
- 所有特征统一归一化以匹配 LR 假设。

## 4 模型结构分析

### 4.1 总统结构框架



#### 1. 输入层

- 输入文本首先通过 BERT Tokenizer 进行分词
- 生成 token IDs 和 attention mask 两个关键张量
- 同时对原始文本进行分词统计，获取长度特征

#### 2. BERT特征提取

- 使用预训练的 BERT-base-uncased 模型
- 12层 Transformer 编码器串联
- 提取最后一层 [CLS] token 的隐藏状态作为句子表示
- 输出维度为 768

#### 3. 特征融合

- 将 BERT 的 [CLS] 向量(768维)与句子长度特征(1维)拼接
- 得到最终的 769 维特征向量

#### 4. 分类器

- 使用 Spark ML 的 LogisticRegression 实现
- 支持 L1/L2 正则化
- 通过 Softmax 输出二分类概率

#### 5. 训练优化

- 3折交叉验证确保模型稳定性
- 网格搜索优化超参数
- 支持 ElasticNet 混合正则化

## 4.2 模型组件说明

BERT (Bidirectional Encoder Representations from Transformers) :

输入处理:

- 词嵌入层: 将 token ID 转换为 768 维向量
- 位置编码: 添加位置信息

Transformer层:

- 12个相同的 Transformer 编码器层
- 每层包含:
- 12头自注意力机制
- 残差连接和层归一化
- 前馈神经网络

分类器结构:

- 输入: 769维特征向量
- 权重矩阵:  $769 \times 2$
- 偏置项: 2维
- Softmax激活输出概率

关键配置:

- 隐藏层维度: 768
- 注意力头数: 12
- Transformer层数: 12

逻辑回归 (Logistic Regression) :

- CoLA 任务 (语法接受性) :

参数网格更大, 包含:

- L2 正则化系数: [0.01, 0.1]
- ElasticNet 混合参数: [0.0, 0.5]
- 最大迭代次数: [50, 100]
- 使用 3 折交叉验证

- SST-2 任务 (情感分析) :

参数网格较简单:

- 仅调整 L2 正则化系数: [0.01, 0.1]
- 固定最大迭代次数: 100
- 同样使用 3 折交叉验证
- 任务相对简单, 参数空间可以更小

## 4.3 其他优化细节

- 使用 joblib 持久化最佳模型系数: 避免每次推理重新训练;
- 添加 length 特征弥补 BERT 对句法长度的弱敏感性;
- 所有 Spark 表达式使用 cache() 减少数据重复计算。

## 5 数据处理与实验设置

### 5.1 数据预处理

CoLA 和 SST-2 数据集均以 TSV 形式提供：

train.tsv: 训练集; dev.tsv: 验证集; test.tsv: 无标签测试集。使用 Pandas 和 PySpark 完成统一格式转换，字段仅保留 sentence 和 label，最终保存为 processed\_data/{task}\_\*.tsv。

### 5.2 内存优化

BERT 特征提取：

- 使用 `torch.no_grad()` 上下文管理器
- 批处理机制 (`batch_size=32`) 减少内存占用
- 及时将 GPU 张量转移到 CPU 并转换为 NumPy 数组

Spark 配置：

- 使用 6GB 驱动器内存
- 设置本地多线程模式 `local[*]`


## 6 实验结果与分析

### 6.1 CoLA 任务结果

验证集指标如下：

- 准确率：0.7680
- 精确率：0.7719
- 召回率：0.9431
- F1分数：0.8489
- Matthews指数：0.4067

```
加载 BERT 模型与 Tokenizer from ./bert_model...
任务: COLA | 使用设备: cuda
加载验证集: processed_data/cola_dev.tsv
```

 验证集评估指标：

```
准确率: 0.7680
精确率: 0.7719
召回率: 0.9431
F1 分数: 0.8489
MCC: 0.4067
```

分类报告：

	precision	recall	f1-score	support
0	0.7469	0.3758	0.5000	322
1	0.7719	0.9431	0.8489	721
accuracy			0.7680	1043
macro avg	0.7594	0.6595	0.6745	1043
weighted avg	0.7642	0.7680	0.7412	1043

```
加载待预测文本: processed_data/cola_test.tsv
预测结果保存到: cola_test_pred.tsv
```

## 6.2 SST-2 任务结果

验证集指标如下：

- 准确率：0.8601
- 精确率：0.8531
- 召回率：0.8761
- F1分数：0.8644
- Matthews指数：0.7202

```
加载 BERT 模型与 Tokenizer from ./bert_model...  
任务: SST2 | 使用设备: cuda  
加载验证集: processed_data/sst2_dev.tsv
```

📊 验证集评估指标：

```
准确率: 0.8601  
精确率: 0.8531  
召回率: 0.8761  
F1 分数: 0.8644  
MCC: 0.7202
```

分类报告：

	precision	recall	f1-score	support
0	0.8678	0.8435	0.8555	428
1	0.8531	0.8761	0.8644	444
accuracy			0.8601	872
macro avg	0.8604	0.8598	0.8599	872
weighted avg	0.8603	0.8601	0.8600	872

```
加载待预测文本: processed_data/sst2_test.tsv  
预测结果保存到: sst2_test_pred.tsv
```

## 6.3 效果分析

- 原始版本中仅使用 BERT + LR 且未调参时，CoLA 任务 MCC 仅为 0.22 左右；
- 添加 length 特征、进行特征标准化和正则调参后提升至 0.4067，增幅达 +18%；
- SST-2 原本准确率仅 80%左右，优化后达到 86.01%。

## 7 结论与展望

本文通过构建 BERT 特征提取器与 PySpark 中的逻辑回归分类器，成功实现了对 CoLA 与 SST-2 两个文本分类任务的建模与评估，取得了令人满意的结果。

优点：

- 利用预训练模型提取深层语义特征，增强泛化能力；
- 结合分布式计算加速训练与特征处理；
- 参数少，部署成本低，适合轻量场景。

局限性：

- 仍依赖 BERT 模型加载，推理耗时相较传统方法较高；
- 不支持多类别或多标签扩展，需进一步改造；
- 无法端到端训练，存在误差传播风险。

展望：

- 尝试使用 BERT 其他层的输出特征
- 添加更多语言学特征（如词性、句法树等）



- 尝试其他预训练模型（如 RoBERTa、ALBERT 等）
- 研究不同的分类器组合（如 SVM、随机森林等）
- 实现模型的增量训练
- 添加文本预处理和后处理模块

## 参考文献

- [1] 胡健. 基于Spark的中文文本情感分析研究[D]. 景德镇陶瓷大学, 2023. DOI:10.27191/d.cnki.gjdtc.2023.000089.
- [2] 崔金英, 周芸竹. 基于BERT模型的文献自动分类研究[J]. 数字与缩微影像, 2025, (02):1-4.
- [3] 李卓冉. 逻辑回归方法原理与应用[J]. 中国战略新兴产业, 2017, (28):114-115. DOI:10.19474/j.cnki.10-1156/f.001686.