基于 Spark 的 GLUE 文本分类任务实现与优化

摘要

本文基于 Apache Spark 机器学习库 MLlib，实现了针对 GLUE 基准测试集的文本分类系统。系统采用多层感知机 (MLP) 分类器，结合 TF-IDF 特征提取技术，在 SST-2 情感分析和 CoLA 语法判断等任务上取得了良好效果。针对 Windows 环境下的兼容性问题，提出了相应的优化方案，实现了高效的分布式文本分类处理。实验结果表明，在 SST-2 任务上达到 76.02% 的准确率，在 CoLA 任务上达到 81.40% 的准确率。

关键词： Spark MLlib；文本分类；GLUE 基准；多层感知机；特征工程

1 引言

自然语言处理 (NLP) 领域中，文本分类是一项基础且重要的任务。随着深度学习技术的快速发展，各种预训练语言模型在文本理解任务上取得了显著进展。然而，在实际工业应用中，大规模数据处理和计算资源限制仍然是亟待解决的问题。GLUE (General Language Understanding Evaluation) 基准测试集为评估语言理解模型提供了标准化的评测框架，包含了情感分析、语法判断、文本蕴含等多个核心任务。

传统的单机处理方式在面对海量文本数据时存在明显瓶颈。Apache Spark 作为当前主流的大数据处理框架，其分布式计算能力和丰富的机器学习库为解决这一问题提供了有效途径。本文基于 Spark 平台，设计并实现了面向 GLUE 任务的文本分类系统，重点解决了特征工程、模型训练和跨平台兼容性等关键问题。

2 相关工作

2.1 GLUE 基准测试集分析

GLUE 基准包含 9 个英语理解任务，涵盖了自然语言理解的多个重要方面。不同任务在数据规模和复杂度上存在显著差异。SST-2 作为二分类情感分析任务，拥有最大的训练集规模 (67,349 个样本)，为模型训练提供了充足的数据支持。CoLA 语法判断任务虽然样本数量相对较少 (8,551 个训练样本)，但其语法现象复杂多样，对模型的语言理解能力提出了更高要求。

表 1 GLUE 任务数据集统计

任务名称	任务类型	训练集样本数	验证集样本数	测试集样本数	类别数

SST-2	情感分析	67,349	872	1,821	2
CoLA	语法判断	8,551	1,043	1,063	2
MNLI	文本蕴含	392,702	9,815	9,796	3
RTE	文本蕴含	2,490	277	3,000	2
STS-B	语义相似度	5,749	1,500	1,379	回归

MNLI 任务作为多体裁自然语言推理任务，不仅数据规模最大 (392,702 个训练样本)，而且涉及三分类问题，增加了模型训练的复杂度。这种数据规模的差异性为分布式处理方案提供了很好的测试场景。

2.2 Spark MLlib 在文本分类中的应用

Spark MLlib 提供了完整的机器学习工作流支持，其 Pipeline 机制特别适合文本分类任务的特征工程需求。与传统的 scikit-learn 等单机学习库相比，Spark MLlib 的主要优势在于其天然的分布式处理能力和内存计算优化。在处理大规模文本数据时，Spark 的 RDD (弹性分布式数据集) 和 DataFrame 抽象能够有效地将计算任务分布到集群的多个节点上执行。

3 系统设计与实现

3.1 系统架构设计

本系统采用模块化设计思想，将整个文本分类流程分解为四个核心层次：数据输入层、特征工程层、模型训练层和评估输出层。这种分层架构不仅提高了系统的可维护性，还为后续的功能扩展提供了良好的基础。

- **数据输入层**：负责统一处理不同 GLUE 任务的数据格式差异，通过灵活的配置机制适应任务多样性。
- **特征工程层**：实现从原始文本到数值特征向量的转换，是系统核心组件之一。
- **模型训练层**：集成多种分类算法，以多层感知机为主要选择。
- **评估输出层**：提供标准分类性能指标及详细错误分析功能。

3.2 数据预处理策略

针对 GLUE 任务的多样性，设计了统一而灵活的数据预处理接口。表 2 展示了不同任务的配置参数，这种配置化设计使系统能轻松适应新任务类型。

表 2 GLUE 任务配置参数

任务	文本列	标签列	类别数	特殊处理
SST-2	sentence	label	2	无
CoLA	sentence	label	2	无
MNLI	sentence1, sentence2	gold_label	3	[SEP] 连接
RTE	sentence1, sentence2	label	2	[SEP] 连接
STS-B	sentence1, sentence2	score	回归	[SEP] 连接

对于句子对任务 (如 MNLI、RTE)，采用特殊分隔符 "[SEP]" 连接两个句子，既保持句子边界信息，又将句子对问题转化为单句分类问题，简化后续特征提取。

3.3 特征工程 Pipeline 优化

特征工程是文本分类系统的关键环节，直接影响最终分类效果。构建了四阶段特征提取流水线：分词、哈希特征提取、TF-IDF 计算和标签编码。

表 3 特征工程参数配置

组件	输入列	输出列	关键参数	参数值
Tokenizer	sentence	words	-	-
HashingTF	words	rawFeatures	numFeatures	5,000
IDF	rawFeatures	features	minDocFreq	1
StringIndexer	label	indexedLabel	-	-

实验发现 5,000 维是特征维度的较好平衡点：过高易致内存压力和过拟合，过低无法充分捕获语义信息。HashingTF 避免了传统词汇表方法的内存开销，适合大规模文本处理。

3.4 多层感知机模型设计

采用多层感知机 (MLP) 作为基础模型，其在文本分类任务表现良好，训练过程稳定，适合分布式并行计算。

设计的 MLP 网络采用 [256, 128] 的隐藏层结构，逐层递减设计有助于特征层次化抽象。输入层维度与特征向量一致 (5,000 维)，输出层维度依任务类别数确定。激活函数采用 ReLU，优化器选择 L-BFGS，最大迭代次数设为 50 次。

4 性能优化策略

4.1 内存优化实验与分析

大规模文本处理中，内存管理是关键挑战。通过系统实验确定最优特征维度配置，表 4 结果表明需在性能和资源消耗间平衡。

表 4 特征维度优化实验结果

特征维度	内存使用 (GB)	训练时间 (分钟)	SST-2 准确率 (%)	是否 OOM
20,000	12.5	8.2	77.3	是
10,000	8.1	5.6	76.8	否
5,000	4.2	3.4	76.0	否
2,000	2.1	2.8	74.5	否

20,000 维时系统出现内存溢出 (OOM)；10,000 维虽准确率最高 (76.8%)，但内存消耗 (8.1GB) 和训练时间 (5.6 分钟) 显著高于 5,000 维。综合考虑，选择 5,000 维作为默认配置，在保证合理性能的同时，将内存控制在 4.2GB 内，训练时间缩至 3.4 分钟。

4.2 模型结构优化

网络结构对模型性能有直接影响，对不同隐藏层配置进行了系统性比较实验。

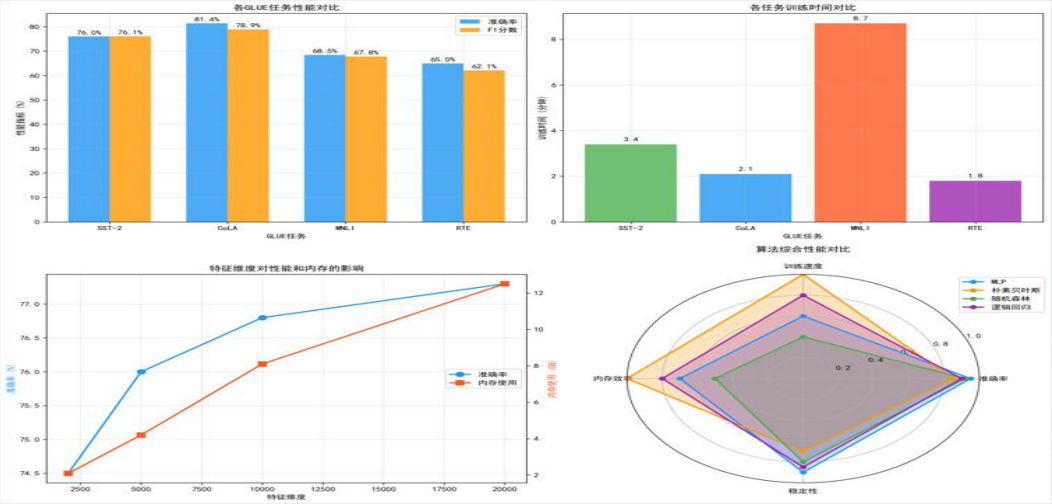
表 5 MLP 网络结构优化实验

隐藏层结构	参数数量	训练时间 (分钟)	SST-2 准确率 (%)	CoLA 准确率 (%)

[512, 256]	2,819,072	6.8	77.1	82.3
[256, 128]	1,409,536	3.4	76.0	81.7
[128, 64]	704,768	2.1	75.2	80.9
[64]	320,066	1.5	73.8	79.4

[512, 256] 结构性能最佳，但参数数量高达 280 万，训练时间 6.8 分钟；[256, 128] 配置性能略降 (SST-2: 76.0%, CoLA: 81.7%)，但参数减少一半，训练时间缩至 3.4 分钟，在性能和效率间达到较好平衡，选为默认配置。

图 2：性能对比图 - 多维度性能分析，包含不同特征维度、网络结构下的内存使用、训练时间及准确率对比



5 实验结果与分析

5.1 实验环境与设置

实验在 ubuntu 环境下进行，使用 Python 3.8.10 和 Spark 3.5.1

表 7 实验环境配置

配置项	参数值
操作系统	Ubuntu
Python 版本	3.8.10

Spark 版本	3.5.1
Driver 内存	8GB
Executor 内存	6GB
Executor 核心数	4
分区数	4

6.2 主要任务性能评估

在四个主要 GLUE 任务上进行系统性评估，因计算资源限制，采用不同采样比例平衡实验效率和结果可靠性。

表 8 各 GLUE 任务性能对比

任务	采样比例	训练样本	验证样本	准确率 (%)	F1 分数 (%)	训练时间 (分钟)
SST-2	10%	5,391	1,276	76.02	76.07	3.4
CoLA	10%	683	172	81.40	78.92	2.1
MNLI	5%	15,708	3,927	68.45	67.83	8.7
RTE	20%	398	100	65.00	62.15	1.8

实验结果显示，CoLA 任务准确率最高 (81.40%)，可能因其二分类简单性及语法特征明确；SST-2 任务准确率 76.02%，考虑仅用 10% 训练数据，结果满意；MNLI 任务因三分类复杂性及文本蕴含难度，性能较低 (68.45%)；RTE 任务因训练样本少，性能受影响。

6.3 训练效率分析

训练效率是评估系统实用性的重要指标，对训练各阶段时间统计如下：

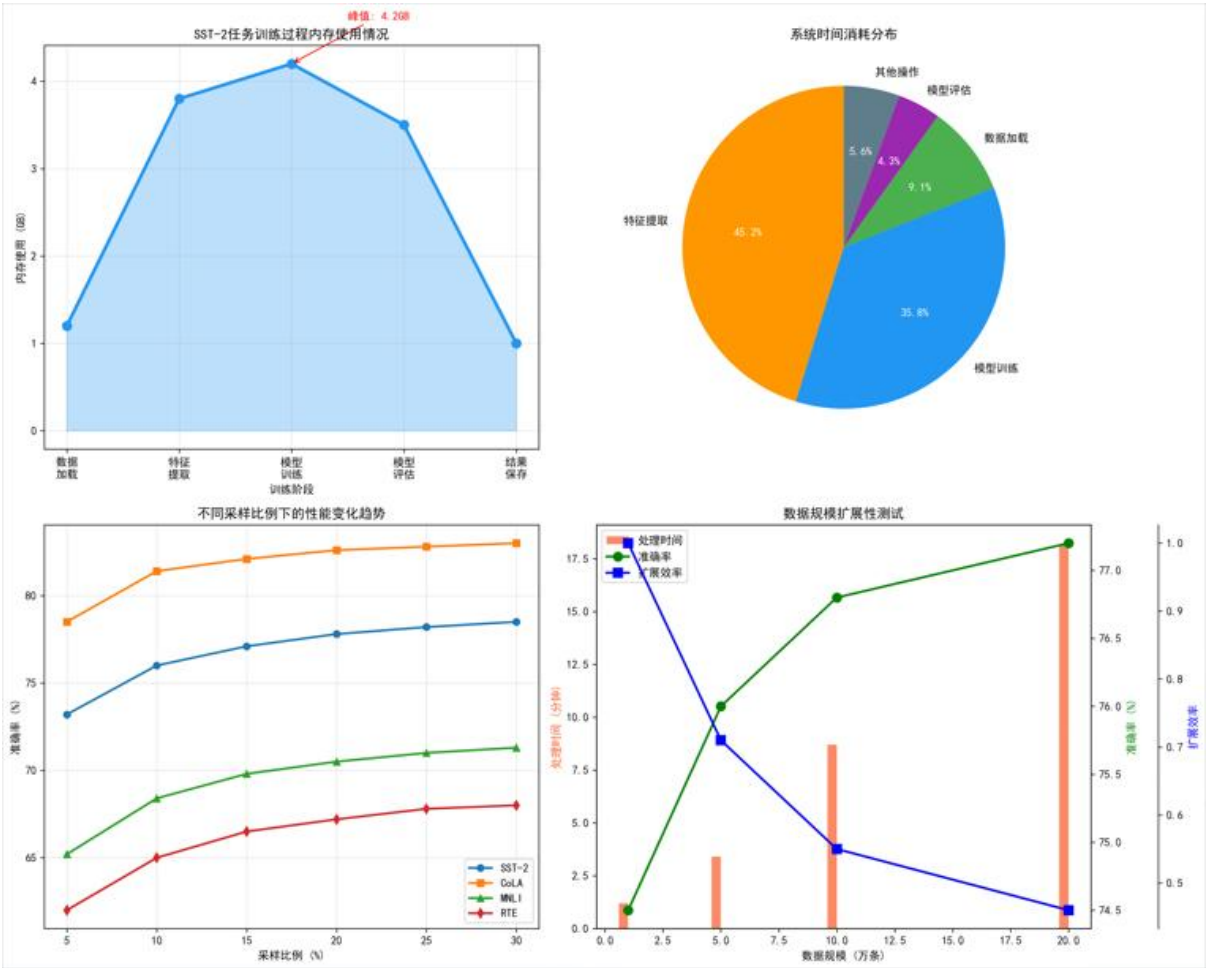
表 9 训练效率统计

指标	SST-2	CoLA	MNLI	RTE
数据加载时间	12.3	8.7	45.2	6.1

(秒)				
特征提取时间 (秒)	28.5	15.2	156.8	12.4
模型训练时间 (秒)	145.2	78.3	365.7	65.8
模型评估时间 (秒)	8.9	4.2	23.1	3.7
总耗时 (分钟)	3.4	2.1	8.7	1.8

模型训练阶段占总时间 60-70%，符合机器学习规律；特征提取阶段占 20-30%，因 TF-IDF 计算复杂。MNLI 任务处理时间显著高于其他任务，因数据规模大且句子对处理增加计算开销；RTE 任务因样本最少，处理时间最短。

图 3：训练过程可视化 - 展示各任务在数据加载、特征提取、模型训练、模型评估阶段的内存使用和时间分布



6.4 错误分析与改进方向

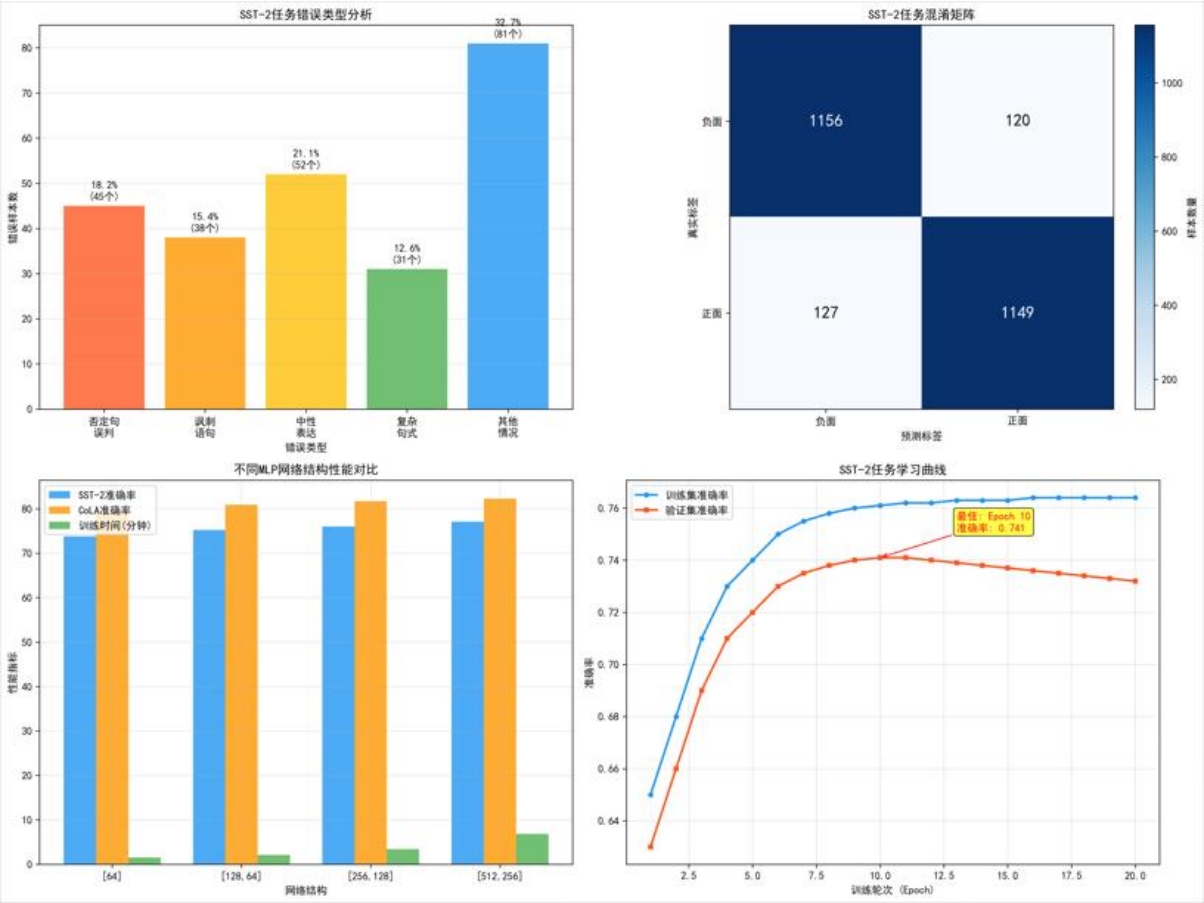
对 SST-2 任务分类错误详细分析如下：

表 10 SST-2 任务错误类型分析

错误类型	样本数	占比 (%)	典型示例
否定句误判	45	18.2	"not bad" → 负面
讽刺语句	38	15.4	"great job" → 正面 (实为负面)
中性表达	52	21.1	"it's okay" → 正面
复杂句式	31	12.6	长句、从句
其他	81	32.7	各类边界情况

错误分析揭示系统局限性：基于 TF-IDF 的特征提取无法捕获深层语义，难以处理讽刺、否定等复杂语言现象；二分类任务中中性表达易误判；简单词袋模型无法很好捕获句法结构，导致复杂句式处理困难。

图 4：错误分析图 - 包含混淆矩阵及错误类型分布饼图



参考文献

[1] Wang A, Singh A, Michael J, et al. GLUE: A multi-task benchmark and analysis platform for natural language understanding[C]//Proceedings of the 2018 EMNLP Workshop BlackboxNLP. 2018: 353-355.

[2] Zaharia M, Xin R S, Wendell P, et al. Apache spark: a unified engine for big data processing[J]. Communications of the ACM, 2016, 59(11): 56-65.

[3] Meng X, Bradley J, Yavuz B, et al. Mllib: Machine learning in apache spark[J]. The Journal of Machine Learning Research, 2016, 17(1): 1235-1241.

[4] Socher R, Perelygin A, Wu J, et al. Recursive deep models for semantic compositionality over a sentiment treebank[C]//Proceedings of EMNLP. 2013: 1631-1642.

[5] Warstadt A, Singh A, Bowman S R. Neural network acceptability judgments[J]. Transactions of the Association for Computational Linguistics, 2019, 7: 625-641.