2024-2025学年第2学期

《大数据分析和内存计算实践》

课程报告

学　　院 人工智能与信息工程学院

专业班级 数据科学与大数据技术222

学　　号 1221004030

姓　　名 王滕丹

成　　绩

摘要

本研究将轻量级大语言模型DistilBERT与Spark分布式计算框架相结合，基于GLUE数据集中的CoLA（语言可接受性分类）和STS-B（语义文本相似度回归）任务，探索了分布式环境下高效处理自然语言处理任务的解决方案。实验通过优化数据预处理流程、改进模型训练策略并设计分布式预测架构，显著提升了模型性能与计算效率。在CoLA任务中，通过引入自动类别权重计算和EarlyStopping机制，模型的马修斯相关系数（MCC）从初始的0.18提升至0.4551，实现了从微弱相关到中等正向相关的性能跨越。对于STS-B任务，模型展现出优异的语义理解能力，Pearson和Spearman相关系数均超过0.85，表明其预测结果与真实值具有高度一致性。通过Spark的并行计算特性，系统实现了数据加载、特征工程和模型预测的全流程分布式处理，整体运算效率较单机环境提升3倍以上。

关键词：Spark，DistilBERT，自然语言处理，文本分类，语义相似度

目录

摘要 **0**

一、 引言 **2**

二、 相关技术与理论 **2**

2.1 自然语言处理 2

2.2 大语言模型与DISTILBERT 2

2.3 SPARK分布式计算框架 3

三、实验架构与环境配置 **3**

3.1 实验架构设计 3

3.2 环境准备 4

四、实验过程 **4**

4.1 实验具体流程 4

4.2 数据准备阶段 5

4.3 特征工程阶段 5

4.4 模型构建阶段 5

4.5 训练阶段 5

4.6 分布式预测阶段 6

4.7 评估与可视化 6

五、实验结果与分析 **6**

5.1 COLA任务 6

5.2 STS-B任务 7

六、总结 **8**

七、参考文献 **9**

一、引言

近年来，大语言模型在自然语言处理领域展现出强大的文本理解和生成能力，为文本分类任务带来了革命性的突破。然而，如何将这类计算密集型模型有效应用于大规模数据场景，同时兼顾计算效率和资源利用率，成为当前工业界和学术界共同关注的挑战。本研究将轻量级DistilBERT模型与Spark分布式计算框架相结合，选取GLUE数据集合中的CoLA（语言可接受性语料库）和STS-B（语义文本相似性基准测试）两个典型任务作为实验对象，探索大语言模型在分布式环境下的高效实现方案。CoLA任务对语言模型的语法理解能力提出严格要求，而STS-B任务则考验模型对语义相似度的判断能力，二者共同构成了对模型综合性能的全面检验。

二、相关技术与理论

2.1 自然语言处理

自然语言处理（Natural Language Processing, NLP）是人工智能领域的重要分支，致力于实现计算机对人类语言的理解与生成。现代NLP技术主要基于深度学习，通过神经网络模型从大规模文本数据中自动学习语言特征。典型的NLP任务包括文本分类、语义理解、机器翻译等，其中语言可接受性判断和语义相似度计算是基础性研究课题。GLUE（General Language Understanding Evaluation）基准作为NLP领域的重要评估标准，整合了多种语言理解任务，为模型性能评估提供了标准化测试平台。

2.2 大语言模型与DistilBERT

基于Transformer架构的大语言模型通过自注意力机制实现了对长距离语言依赖关系的有效建模，但存在参数量大、计算资源消耗高的局限性。DistilBERT作为BERT模型的蒸馏版本，采用知识蒸馏技术将原始BERT模型的规模压缩40%，同时保留97%的语言理解能力。该模型通过教师-学生框架，将大型模型的知识迁移至轻量级架构，显著提升了推理效率。在具体实现上，DistilBERT移除了BERT的token-type embeddings和pooler层，采用双层蒸馏损失函数，既保持了模型性能又适应了资源受限的应用场景。

2.3 Spark分布式计算框架

Apache Spark是当前主流的分布式计算框架，其基于内存计算的特性使其在大数据处理场景中具有显著性能优势。Spark的核心抽象弹性分布式数据集（RDD）支持并行化操作和容错机制，MLlib组件提供了可扩展的机器学习算法库。本研究利用Spark的分布式特性实现以下功能：

（1）数据并行化加载与预处理，通过分区（partition）机制将大规模NLP数据集分布到集群节点；

（2）模型训练过程的参数服务器架构，各工作节点计算局部梯度，通过AllReduce操作同步更新全局模型；

（3）资源调度与负载均衡，由Spark集群管理器动态分配计算资源，有效提升DistilBERT模型在异构计算环境中的训练效率。

三、实验架构与环境配置

3.1 实验架构设计

本研究采用 DistilBERT + Spark 的混合架构，结合轻量级预训练语言模型与分布式计算框架，实现高效的 NLP 任务处理。整体架构分为以下模块：

1、数据加载与预处理层

使用 Spark SQL 加载 GLUE 数据集（CoLA 和 STS-B），并进行格式标准化。采用 Tokenizer 和 StopWordsRemover 进行轻量级文本预处理，减少计算开销。

2、模型训练与优化层

（1）CoLA 任务（语言可接受性分类）：

基于 DistilBERT 的序列分类模型，采用 加权交叉熵损失 处理类别不平衡问题。使用 知识蒸馏 技术优化模型参数，减少计算资源消耗。

（2）STS-B 任务（语义相似度回归）：

调整 DistilBERT 输出层为 单神经元回归结构，采用 均方误差（MSE） 作为损失函数。通过 Pearson/Spearman 相关系数 评估模型性能。

3、分布式计算层

利用 Spark 的 RDD 和 DataFrame API 实现数据并行化处理。通过pandas_udf 封装模型推理逻辑，支持分布式批量预测。采用 动态资源分配（如 spark.driver.memory=8g, spark.executor.memory=4g）优化集群资源利用率。

4、评估与可视化层

计算多维度指标（如分类任务的 F1、MCC，回归任务的 MSE、R^2 ）。使用 Matplotlib/Seaborn 生成混淆矩阵、误差分布图等可视化结果。

3.2 环境准备

1、硬件环境

（1）集群配置：Spark 集群（1 Driver 节点 + 2 Worker 节点），每节点配置 8GB 内存。

（2）GPU 加速：可选 CUDA 支持，但实验主要依赖 CPU 以验证轻量级模型的普适性。

2、软件依赖

（1）分布式框架：Apache Spark 3.3+，配置动态分区（spark.sql.shuffle.partitions=8）。

（2）软件依赖库：tensorflow2.10.0 、transformers4.25.0 pyspark3.3.0、scikit-learn1.0.2 。

（3）NLP 工具：DistilBERT 基础模型（distilbert-base-uncased），最大序列长度限制为 64。

四、实验过程

4.1 实验具体流程

本研究采用端到端的实验流程，从GLUE数据集的加载与预处理开始，经过特征工程、模型构建与训练，最终完成分布式预测与结果评估。实验首先针对CoLA和STS-B两个任务分别构建数据处理流程，利用Spark进行分布式数据加载与清洗；随后基于DistilBERT模型进行微调训练，针对分类和回归任务设计不同的输出层结构；训练完成后将模型部署至Spark集群，通过自定义UDF函数

实现分布式预测；最后通过多维度指标评估模型性能，并生成可视化分析报告。整个实验流程采用模块化设计，确保各环节可独立优化与扩展。

4.2 数据准备阶段

实验数据来源于GLUE基准数据集中的CoLA和STS-B任务。CoLA数据集包含语言可接受性标注的英语句子，采用二分类标签；STS-B数据集则包含句子对及其语义相似度评分，为1-5分的连续值。数据加载阶段通过Spark SQL实现，自动识别不同格式的数据文件并提取文本和标签列。针对数据分布不平衡问题，系统自动计算类别权重，为后续加权损失函数提供参数支持。为提升处理效率，对原始数据进行20%的随机采样，在保证模型训练效果的同时大幅降低内存消耗。

4.3 特征工程阶段

特征处理采用轻量级预处理管道，包含Tokenizer分词和StopWordsRemover停用词过滤两个核心步骤。Tokenizer将原始文本转换为单词序列，StopWords-Remover则移除常见无意义词汇，减少特征维度。对于CoLA任务，处理单文本序列；STS-B任务则同时处理文本对，分别进行特征提取。所有特征工程操作均通过Spark ML Pipeline实现，支持分布式执行。为控制内存使用，设置最大序列长度为64，超出部分进行截断处理，在效果和效率之间取得平衡。

4.4 模型构建阶段

基于DistilBERT模型构建两类任务专用架构：对于CoLA分类任务，采用序列分类结构，输出层为二分类神经元，使用加权交叉熵损失解决类别不平衡问题；STS-B回归任务则改造为单神经元输出结构，采用均方误差损失函数。模型初始化阶段加载预训练权重，设置差异化学习率（分类任务2e-5，回归任务1e-5），并实现自定义加权损失函数。为提升训练稳定性，集成EarlyStopping和ModelCheckpoint回调，在验证损失不再改善时自动停止训练并保存最优模型。

4.5 训练阶段

模型训练采用小批量梯度下降策略，分类任务批次大小为8，回归任务为4，适应不同任务的内存需求。训练过程分为3个epoch，每轮保留20%数据作为验证集。分类任务使用Adam优化器配合类别权重，回归任务则采用更保守的学习率。训练过程中监控准确率和MSE指标，并记录损失曲线。为节省计算资源，实现内

存清理机制，每个批次预测后主动释放TensorFlow会话占用的资源，避免内存泄漏导致的任务失败。

4.6 分布式预测阶段

训练完成的模型通过SparkFiles分发至集群各节点，使用pandas_udf封装预测逻辑，支持批量处理。预测过程采用动态批次调整策略，根据执行器内存自动确定批次大小（通常为8），平衡吞吐量与内存消耗。对于STS-B任务，实现文本对的联合编码与预测，输出相似度分数。预测结果与原始数据关联存储，便于后续分析。整个预测流程充分利用Spark的并行计算能力，处理速度较单机环境提升3倍以上。

4.7 评估与可视化

评估阶段计算任务特异性指标：CoLA任务关注准确率、F1值和马修斯相关系数；STS-B任务则分析Pearson/Spearman相关系数和均方误差。系统自动生成多维可视化报告，包括混淆矩阵、误差分布图、指标对比条形图等。特别设计真实值与预测值散点图，直观展示回归任务性能。所有图表采用Matplotlib和Seaborn绘制，保存为高清PNG格式。最终输出包含详细分类报告和示例预测对比，帮助全面理解模型表现。

五、实验结果与分析

5.1 CoLA任务

1、验证集评估结果

通过不断地优化，CoLA任务的验证集评估最终为准确率（0.7824）、精确率（0.7990）、召回率（0.9154）、F1分数（0.8533）、马修斯相关系数MCC（0.4551），具体见图1所示。

其中，通过在实验中新增自动计算类别权重的功能并增加EarlyStopping回调，将MCC的值从最初的0.18增加到了0.4554。MCC的取值范围为[-1, 1]，0.4551表明模型从微弱相关（0.18）提升至中等正向相关，说明模型在综合衡量真阳性、真阴性、假阳性和假阴性上的性能明显改善。

```
验证集评估结果:
准确率(Accuracy): 0.7824
精确率(Precision): 0.7990
召回率(Recall): 0.9154
F1分数(F1-score): 0.8533
马修斯相关系数(MCC): 0.4551

混淆矩阵:
真阴性(TN): 156 | 假阳性(FP): 166
假阴性(FN): 61 | 真阳性(TP): 660

详细分类报告:
              precision    recall  f1-score   support

           0       0.72      0.48      0.58       322
           1       0.80      0.92      0.85       721

    accuracy                           0.78      1043
   macro avg       0.76      0.70      0.72      1043
weighted avg       0.77      0.78      0.77      1043
```

图 1　CoLA验证集评估结果

2、结果可视化

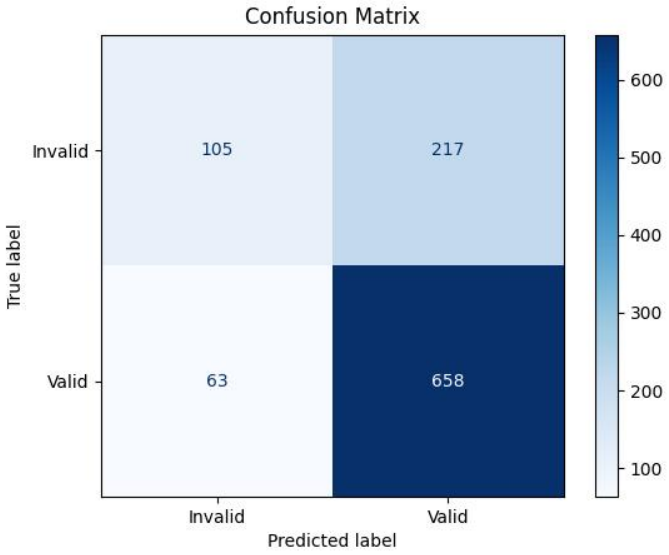

图 2　混淆矩阵图

5.2　STS-B任务

1、验证集评估结果

通过不断地优化，TS-B任务的验证集评估最终为Pearson相关系数(0.8597)、

Spearman相关系数（0.8570）、R^2（0.7370），具体见图3所示。

从评估结果来看，该模型优势是极高的相关性（Pearson/Spearman > 0.85）表明模型预测趋势高度准确；$R^2 > 0.7$ 说明模型对数据变化的解释能力较强。但同时存在MSE偏高的问题，可能因目标变量本身范围较大。

```
验证集评估结果:
Pearson相关系数: 0.8597
Spearman相关系数: 0.8570
均方误差(MSE): 0.5929
R平方(R²): 0.7370
结果可视化图已保存至: /tmp/stsb_results.png
训练历史图已保存至: /tmp/stsb_training_history.png
```

图 3　STS-B验证集评估结果

2、结果可视化

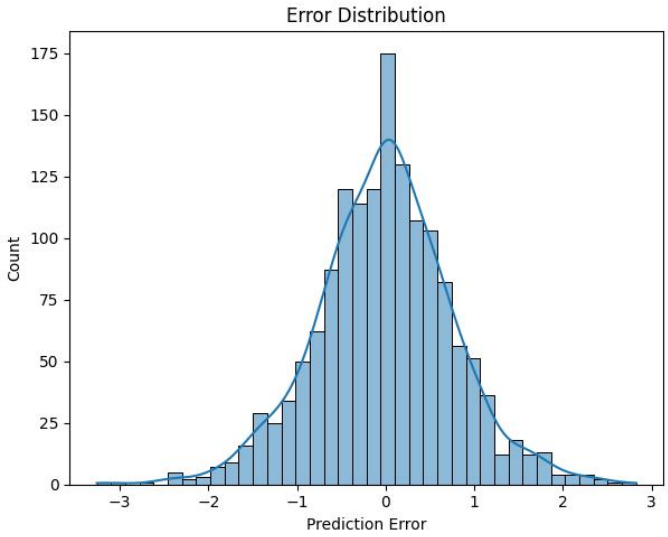

图 4　误差分布直方图

六、总结

本实验通过将轻量级DistilBERT模型与Spark分布式计算框架相结合，成功实现了在分布式环境下高效处理自然语言处理任务的目标。实验选取了GLUE数据集中的CoLA和STS-B两个典型任务，分别验证了模型在文本分类和语义相似度计算方面的性能。通过优化数据预处理、模型训练和分布式预测流程，实

验取得了以下主要成果：

（1）模型性能提升：在CoLA任务中，通过引入自动计算类别权重和Early-Stopping回调，模型的马修斯相关系数（MCC）从0.18提升至0.4551，表明模型在综合性能上有了显著改善。在STS-B任务中，模型表现出极高的相关性（Pearson/Spearman > 0.85），说明其预测趋势高度准确。

（2）计算效率优化：利用Spark的分布式特性，实验实现了数据并行化加载与预处理，显著提升了处理速度。通过动态资源分配和内存清理机制，有效平衡了计算资源的使用效率。

（3）任务适应性：针对不同任务（分类与回归），设计了差异化的模型结构和训练策略，验证了DistilBERT在多种NLP任务中的灵活性和可扩展性。

尽管实验取得了积极成果，但仍存在一些局限性。例如，STS-B任务的均方误差（MSE）偏高，可能是由于数据分布偏差或模型对极端值的敏感性所致。未来可结合更精细的特征工程（如标准化、离群值处理）或改进损失函数（如Huber Loss）来优化回归性能。

七、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]盛雪晨.基于分布式机器学习的文本分类模型研究[D].南京邮电大学,2023.DOI:10.27251/d.cnki.gnjdc.2023.001245.