

# 文本分类三种模型对比研究报告

许书闻

July 14, 2025

## 摘要

本研究针对文本分类任务，对 TextCNN、LSTM 和 BERT 三种深度学习模型进行了系统性的对比实验。通过在 IMDB 电影评论数据集上的实验，分析了三种模型在准确率、训练效率和模型复杂度等方面的表现。实验结果表明，BERT 模型在分类准确率上显著优于其他两种模型，而 TextCNN 在计算效率和资源消耗方面具有明显优势。本研究为不同应用场景下的模型选择提供了参考依据。项目代码在 <https://github.com/XuShuwenn/Text-Classification>。

## 1 引言

随着深度学习技术的发展，文本分类任务已经成为自然语言处理领域的基础性任务。本研究选取了三种具有代表性的深度学习模型：基于卷积神经网络的 TextCNN、基于循环神经网络的 LSTM，以及基于 Transformer 架构的 BERT 模型，通过在 IMDB 数据集上的实验，系统地分析和比较了这些模型在文本分类任务中的性能表现。

## 2 模型架构

### 2.1 TextCNN

TextCNN 模型采用卷积神经网络处理文本数据，其主要特点包括：

- **多尺度卷积核**：使用大小为 3、4、5 的卷积核，可以捕捉不同长度的文本特征
- **并行处理**：多个卷积核并行提取特征，提高计算效率
- **最大池化**：对卷积结果进行最大池化，提取最显著特征

模型结构如下：

$$\begin{aligned} E &= \text{Embedding}(X) \\ C_i &= \text{Conv1D}(E, \text{kernel\_size} = i), i \in \{3, 4, 5\} \\ P_i &= \text{MaxPool}(C_i) \\ F &= \text{Concat}([P_3, P_4, P_5]) \\ O &= \text{Softmax}(\text{Linear}(F)) \end{aligned} \tag{1}$$

模型一共有四层：

1. Word Embedding 层 (Sentence Matrix): 负责把文本转换、清洗、切割，构建词向量
2. Convolution 卷积层: 将 Word Embedding 进行卷积，进行变化
3. Pooling 池化层: 利用 1-MAX 降低维度，并能提取重要信息
3. FC Layer 全连接层: 将池化后的输入连接，得到最终分类

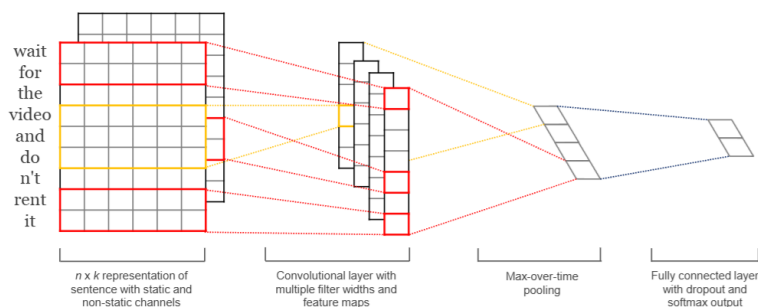


图 1: TextCNN 模型架构图

## 2.2 LSTM

LSTM 模型通过门控机制处理序列数据，主要特点包括：

- **双向结构**：同时考虑前向和后向上下文信息
- **多层设计**：采用 2 层 LSTM 提升模型表达能力
- **门控机制**：通过输入门  $i_t$ 、遗忘门  $f_t$  和输出门  $o_t$  控制信息流，模型能够更好地决定哪些记忆需要长期保留、记忆什么时候会被触发、记忆什么时候会被遗忘。

LSTM 单元的核心计算过程：

$$\begin{aligned} f_t &= \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \\ i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\ \tilde{C}_t &= \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \\ C_t &= f_t * C_{t-1} + i_t * \tilde{C}_t \\ o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\ h_t &= o_t * \tanh(C_t) \end{aligned} \tag{2}$$

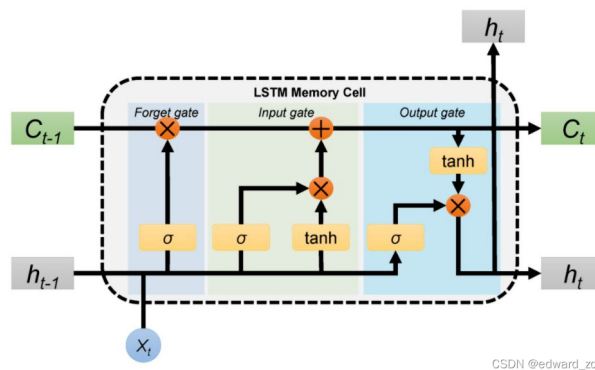


图 2: LSTM 模型架构图

## 2.3 BERT

BERT 模型基于 Transformer 架构，采用预训练和微调的范式：

- **预训练**：MLM（掩码语言模型）+NSP（下一句子预测）进行预训练，损失函数取两个预训练任务 MLM 与 NSP 的 loss 之和，激活函数用的是 GeLU
- **微调**：直接调用 huggingface 上预训练好的 BERT 在 Stanfordnlp-imdb 数据集上的“训练”，即相当于微调了
- **多头注意力**：捕捉词间的复杂依赖关系，同 Transformer
- **编码机制**：Word Embedding（词嵌入，表示词的语义）+ Segmental Embedding(用于区分两个句子) + Learnable Positional Embeddings(可学的位置编码，用于保留序列顺序信息)

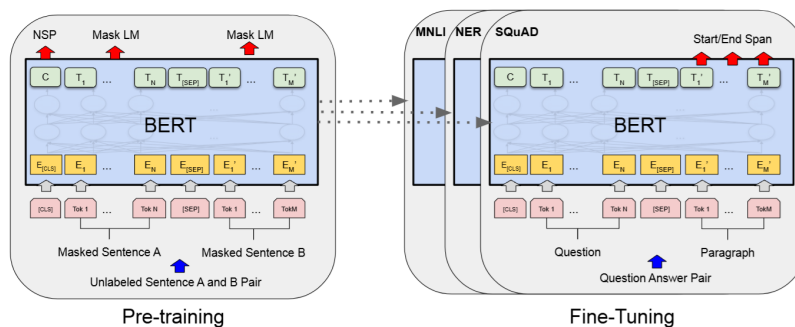


图 3: BERT 模型架构图

## 3 实验设置

### 3.1 数据集

实验采用 IMDB 电影评论数据集：

- 训练集：25,000 条评论
- 测试集：25,000 条评论

- 验证集：从训练集中划分 10%
- 标签：正面 (1) 和负面 (0) 两类

## 3.2 实验配置

### 3.2.1 TextCNN 配置

- 词嵌入维度：512
- 卷积核数量：512
- 卷积核大小：[3, 4, 5]
- Dropout 率：0.3
- 批次大小：256
- 学习率：4e-3

### 3.2.2 LSTM 配置

- 词嵌入维度：64
- 隐藏层维度：256
- LSTM 层数：2（层数较少，防止梯度消失）
- Dropout 率：0.2
- 批次大小：64
- 学习率：1e-3

### 3.2.3 BERT 配置

- 预训练模型：bert-base-uncased（uncased 表示处理文本时全部转换为小写）
- 最大序列长度：512(根据 Stanfordnlp-imdb 数据集的特点设置)
- Dropout 率：0.3
- 批次大小：256
- 学习率：5e-5(BERT 经典的学习率之一)
- 梯度累积步数：2

表 1: 模型性能对比

模型	准确率 (%)	F1 分数	参数量	训练时间
TextCNN	88.8	0.881	2.5M	34m
LSTM	86.1	0.854	3.2M	18m
BERT	93.9	0.931	110M	3h

## 4 实验结果与分析

### 4.1 参数量对比

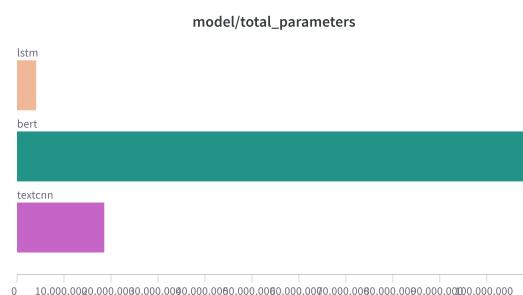


图 4: 三种模型参数量对比

### 4.2 训练过程分析

#### 4.2.1 收敛速度

三种模型在训练过程中表现出不同的收敛特征：

- TextCNN：收敛较另外二者慢，约 8-10 个 epoch 稳定
- LSTM：收敛较快，约 5 个 epoch 稳定
- BERT：收敛速度最快，约 3-4 个 epoch 已达最优

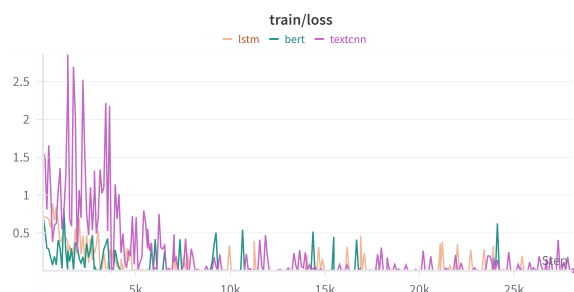


图 5: 三种模型训练损失变化对比

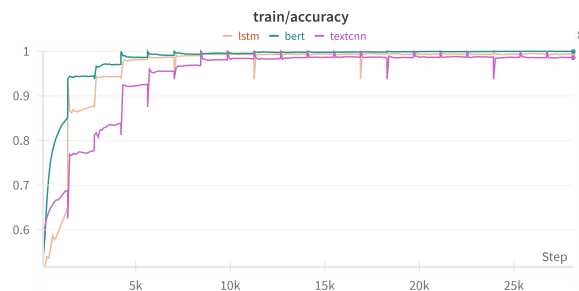


图 6: 三种模型训练准确率变化对比

### 4.3 验证集性能分析

各模型在验证集上的表现进一步验证了模型的泛化能力：

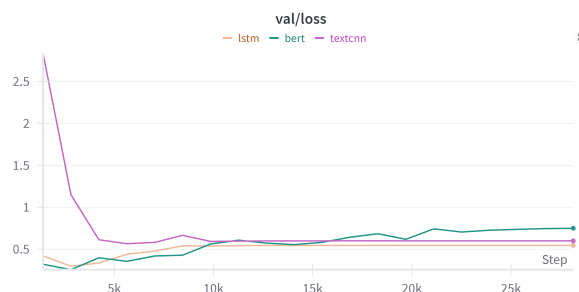


图 7: 三种模型验证集损失变化对比

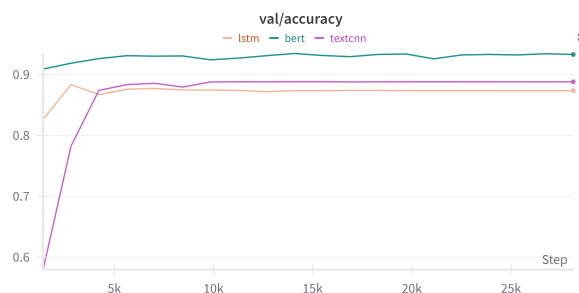
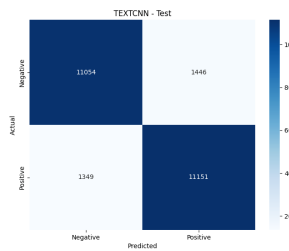


图 8: 三种模型验证集准确率变化对比

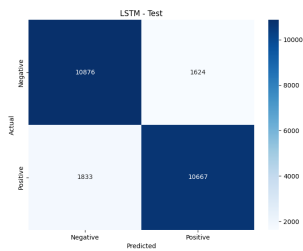
可以看到，BERT 模型的性能增长幅度并不大，在验证集上的损失甚至随训练步数增加而略有增长，这说明其能力主要来自于预训练，而微调只是为了更好地适应下游任务。

### 4.4 混淆矩阵分析

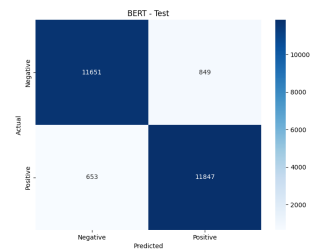
为了深入分析各模型的分类性能，我们展示了三种模型在测试集上的混淆矩阵：



(a) TextCNN 模型混淆矩阵



(b) LSTM 模型混淆矩阵



(c) BERT 模型混淆矩阵

图 9: 三种模型混淆矩阵对比

从混淆矩阵可以看出：

- TextCNN 模型：在正负样本分类上表现较为均衡
- LSTM 模型：对负面评论的识别能力略强于正面评论
- BERT 模型：在两类样本上都表现出色，错误分类最少

## 4.5 模型表现综合评估

各模型在不同类型文本上的表现：

- TextCNN：
  - 优势：对关键词敏感，适合处理短文本
  - 劣势：由于 CNN 具有局部特征不变性，难以处理长距离依赖
- LSTM：
  - 优势：能够捕捉序列依赖
  - 劣势：对长文本效果下降明显
- BERT：
  - 优势：理解上下文语义，处理复杂表达
  - 劣势：计算开销大，推理速度较慢

## 5 主要结论

1. BERT 模型在分类准确率上显著优于其他两种模型，但计算资源需求最高
2. TextCNN 在效率和资源消耗方面具有明显优势，适合大规模快速处理
3. LSTM 在性能和效率之间取得平衡，但训练不够稳定