文本分类实验报告

人工智能专业本科生

April 30, 2025

1 实现细节

参照作业要求，我们实现了 Log-linear 模型和 BERT 模型用于文本分类任务，并在 20-Newsgroups 和 Hallmarks of Cancer Corpus (HoC) 数据集上进行了评估。

1.1 Log-linear 模型

此模型结合了 TF-IDF 特征提取和逻辑回归分类器。

1.1.1 TF-IDF 特征提取

- 使用 1-gram 和 2-gram。
- 最大特征数限制为 50,000。
- 应用 sublinear TF 变换（对频率取对数）。
- 移除了英文停用词。

1.1.2 逻辑回归分类器

- L2 正则化，惩罚系数 C 设置为 4.0。
- 采用 `class_weight='balanced'` 以处理类别不平衡。
- 最大迭代次数设置为 1000，随机种子为 42。

部分关键代码 (来自 **main.py**)

```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.pipeline import Pipeline

clf = Pipeline([
    ('tfidf', TfidfVectorizer(
        ngram_range=(1,2),
        max_features=50000,
        sublinear_tf=True,
        stop_words='english')),
    ('lr', LogisticRegression(
        max_iter=1000,
        C=4.0,
        class_weight='balanced',
        n_jobs=-1,
        random_state=42)) # RNG is 42 in main.py
])
```

1.2 BERT 模型

此模型基于 Hugging Face Transformers 库对 `bert-base-uncased` 模型进行微调。

1.2.1 输入处理

- 文本最大长度设置为 128。
- 使用自动填充（padding）与截断（truncation）。

1.2.2 训练参数

- 训练轮数（Epochs）：3。
- 批处理大小（Batch size）：8。
- 学习率（Learning rate）：2×10^{-5}。
- 随机种子设为 42。

部分关键代码 **(来自 main.py)**

```
from transformers import (BertTokenizerFast, BertForSequenceClassification,
                          TrainingArguments, Trainer)

# Tokenizer and Dataset mapping as in main.py
tokenizer = BertTokenizerFast.from_pretrained('bert-base-uncased')
# ... ds mapping code ...

model = BertForSequenceClassification.from_pretrained(
    'bert-base-uncased', num_labels=num_labels) # num_labels from data loading

training_args = TrainingArguments(
    output_dir='outputs/tmp',
    per_device_train_batch_size=8,
    per_device_eval_batch_size=8,
    num_train_epochs=3,
    learning_rate=2e-5,
    logging_steps=200, # As per report.pdf example
    seed=42 # RNG is 42 in main.py
)

trainer = Trainer(model=model,
                  args=training_args,
                  train_dataset=ds['train'])

trainer.train()
```

2 实验结果

我们在训练集和测试集上使用 Accuracy、Macro-F1 和 Micro-F1 指标评估了模型性能。结果如下表所示（数据来自 results.json）：

3 结果分析与讨论

基于实验结果，我们进行了以下分析：

- 准确率比较：在两个数据集的测试集上，BERT 模型的准确率均优于 Log-linear 模型。尤其是在 HoC 数据集上，BERT 的测试准确率（0.7500）显著高于 Log-linear 模型（0.6333），表明 BERT 在该任务上具有更好的泛化能力。
- 过拟合情况： Log-linear 模型在两个数据集的训练集上均表现出非常高的准确率（均 > 0.96），但在测试集上准确率大幅下降，显示出明显的过拟合现象。相比之下，BERT 模型的训练集和测试集准确率差距较小，泛化能力更强。

Table 1: 模型在不同数据集上的性能表现

模型	数据集	集合	Accuracy	Macro-F1	Micro-F1
Log-linear	20news	训练集	0.9673	0.9701	0.9673
		测试集	0.6949	0.6857	0.6949
Log-linear	HoC	训练集	0.9770	0.9827	0.9770
		测试集	0.6333	0.6064	0.6333
BERT	20news	训练集	0.8964	0.8913	0.8964
		测试集	0.7059	0.6930	0.7059
BERT	HoC	训练集	0.8622	0.7795	0.8622
		测试集	0.7500	0.6119	0.7500

- 数据集特性影响： 20 Newsgroups 数据集样本量较大，Log-linear 模型依赖的 n-gram 特征能够捕捉较丰富的信息，因此取得了相对不错的测试性能。HoC 数据集样本量较少且包含专业术语，Log-linear 模型难以捕捉深层语义，而 BERT 借助预训练知识，在该数据集上表现更优越。
- **Macro-F1** 与 **Micro-F1** 差异：在两个数据集上，Macro-F1 分数均低于 Micro-F1 分数。这在 HoC 数据集上尤为明显（BERT 测试集：Macro-F1 0.6119 vs Micro-F1 0.7500）。这通常表明数据集中存在类别不平衡，模型在样本较少的类别上表现较差，拉低了 Macro-F1 的平均值。
- 模型能力比较： Log-linear 模型依赖表层统计特征，实现简单快速，但在小样本或专业领域数据集上效果有限。BERT 模型能够理解上下文语义，泛化能力和对专业领域任务的适应性更强。