深 圳 大 学 实 验 报 告

课程名称：自然语言处理

实验项目名称：实验 1：对话文本建模与分类

学院：计算机与软件学院

专业：软件工程

指导教师：陈俊扬

报告人：　学号：　班级：　

实验时间：2025 年 11 月 20 日

实验报告提交时间：2025 年 11 月 24 日

教务部制

实验目的与要求：

（1）了解基于对话数据的诈骗检测任务与基本概念
（2）掌握中文对话数据集的预处理与标注格式（如 specific_dialogue_content, interaction_strategy, call_type, is_fraud, fraud_type 等字段）
（3）掌握使用大语言模型或深度学习模型进行对话分类（诈骗/非诈骗）的基本方法
（4）通过编程实践，完成诈骗对话识别实验并统计分类性能指标

本实验通过对话文本的建模与分类，掌握诈骗检测的基本流程。主要包含以下内容：
1. 使用模型（如 SVM、BERT、Qwen、ChatGLM3 或 LLaMA）对中文对话进行编码并输出预测结果（诈骗/非诈骗）
2. 将预测结果与真实标签（is_fraud 字段）进行对比，计算准确率
3. 分别统计诈骗类与非诈骗类的分类准确率，分析模型效果

方法、步骤：

1. 准备数据
    * 使用提供的对话数据集。
    * 提取 `specific_dialogue_content` 作为输入文本，`is_fraud` 作为真实标签（0=非诈骗，1=诈骗）。

2. 安装并运行模型
    * 调用本地或者云端服务器，运行 SVM、BERT、Qwen、ChatGLM3 或 LLaMA。

3. 实验任务
    （1）编写 prompt，调用模型判别每条对话是否为诈骗（输出 0=非诈骗，1=诈骗）
    （2）保存模型预测结果，与训练集和测试集中的真实标签进行比对
    （3）统计以下指标：

    * accuracy = 预测正确的总数量 / 总样本数量`
    * accuracy_fraud = 诈骗样本中预测正确的数量 / 总诈骗样本数量`
    * accuracy_nonfraud = 非诈骗样本中预测正确的数量 / 总非诈骗样本数量`

4. 结果分析
    * 对错误案例进行分析，归纳模型在诈骗检测中的不足

实验过程及内容：
（挑选关键部分，截图运行结果）

实验环境与工具

编程语言：Python 3.x
核心库：
pandas：用于数据读取与处理。
openai：用于调用兼容 OpenAI 协议的大模型 API。
sklearn：用于计算分类性能指标。
模型：Alibaba Cloud Qwen3-max。

①数据准备与预处理

加载数据：使用 pandas 读取 .csv 文件，代码中加入了 try-except 块以兼容 utf-8-sig 和 gbk 两种常见编码，防止中文乱码。

标签标准化：原始数据集中的 is_fraud 字段可能包含 'TRUE', 'FALSE', '1', '0' 等多种格式。代码构建了一个映射字典 map_dict，将所有异构标签统一映射为整数：

1 代表 诈骗

0 代表 非诈骗

```python
def load_and_clean_data(filepath):
    """读取数据并标准化标签"""
    if not os.path.exists(filepath):
        raise FileNotFoundError(f"找不到文件: {filepath}")

    try:
        df = pd.read_csv(filepath, encoding='utf-8-sig')
    except:
        df = pd.read_csv(filepath, encoding='gbk')

    # 标签映射：统一转为 0 和 1
    map_dict = {'TRUE': 1, 'FALSE': 0, True: 1, False: 0, '1': 1, '0': 0, 1: 1, 0: 0}
    df['label_id'] = df['is_fraud'].map(map_dict).fillna(0).astype(int)
    return df
```

②构建提示词 (Prompt Engineering)

角色设定：System Prompt 设定为“反电信诈骗专家”，为模型确立行为基准。

任务描述：明确要求模型“分析对话内容”并“判断是否诈骗”。

约束输出：为了方便代码自动解析结果，Prompt 中强制要求仅输出数字 1 或 0，并禁止输出解释性文字。这避免了后处理时的正则提取困难。

```python
def get_llm_prediction(content):
    """单次调用大模型"""
    prompt = f"""
你是一名反电信诈骗专家。请分析以下对话内容，判断其是否属于电信诈骗。
对话内容：{content}
如果是诈骗，仅输出数字 1；如果不属于诈骗，仅输出数字 0。不要输出其他内容。
"""
    retries = 3
    for i in range(retries):
        try:
            completion = client.chat.completions.create(
                model=Config.MODEL_NAME,
                messages=[
                    {"role": "system", "content": "You are a helpful assistant."},
                    {"role": "user", "content": prompt},
                ],
                stream=False
            )
            res = completion.choices[0].message.content.strip()
            if "1" in res: return 1
            return 0
        except Exception:
            if i == retries - 1: return 0  # 失败默认返回0
            time.sleep(1)
```

③多线程并发推理

由于实验数据量较大，单线程串行调用 API 效率极低。代码引入了 ThreadPoolExecutor 实现并发处理。

并发池：设置 MAX_WORKERS（如 8 线程），同时发送多个 HTTP 请求。

错误重试机制：在 get_llm_prediction 函数中增加了 try-except 和循环重试。如果遇到网络波动或 API 超时，程序会自动等待 1 秒后重试（最多 3 次），大大提高了实验的稳定性（Robustness）。

结果对齐：使用 future_to_index 字典记录每个线程处理的数据索引，确保最终写入 CSV 时，预测结果与原始数据行一一对应，不会乱序。

```python
def process_dataset_multithread(df):
    """多线程处理整个数据集"""
    total = len(df)
    print(f">>> 开始全量处理测试集，共 {total} 条数据...")
    print(f"    模型: {Config.MODEL_NAME} | 线程数: {Config.MAX_WORKERS}")

    results = [0] * total

    with ThreadPoolExecutor(max_workers=Config.MAX_WORKERS) as executor:
        future_to_idx = {
            executor.submit(get_llm_prediction, row['specific_dialogue_content']): idx
            for idx, row in df.iterrows()
        }

        for future in tqdm(as_completed(future_to_idx), total=total, desc="进度"):
            idx = future_to_idx[future]
            try:
                results[idx] = future.result()
            except:
                results[idx] = 0

    df['prediction'] = pd.Series(results, index=df.index)
    return df
```

④指标计算与结果导出

利用 sklearn.metrics.accuracy_score 并配合 Pandas 的布尔索引（Boolean Indexing）分别计算了三个指标：

整体准确率：所有样本预测正确的比例。

诈骗样本准确率 (Recall for Fraud)：df[label==1] 中预测为 1 的比例。这是反诈任务中最关键的指标，代表“由于模型漏报导致的风险程度”。

非诈骗样本准确率 (Recall for Non-Fraud)：df[label==0] 中预测为 0 的比例。代表模型的误报率控制能力。

```python
def calculate_metrics(df):
    """计算准确率指标"""
    y_true = df['label_id'].values
    y_pred = df['prediction'].values

    acc = accuracy_score(y_true, y_pred)

    fraud_df = df[df['label_id'] == 1]
    acc_fraud = accuracy_score(fraud_df['label_id'], fraud_df['prediction']) if len(fraud_df) > 0 else 0.0

    non_fraud_df = df[df['label_id'] == 0]
    acc_nonfraud = accuracy_score(non_fraud_df['label_id'], non_fraud_df['prediction']) if len(
        non_fraud_df) > 0 else 0.0

    print(f"\n========= 核心指标统计 =========")
    print(f"总准确率 (Accuracy):           {acc:.2%}")
    print(f"诈骗样本识别率 (Recall Fraud):   {acc_fraud:.2%}")
    print(f"非诈骗样本识别率 (Recall Non):  {acc_nonfraud:.2%}")
```

数据处理分析（如不涉及可写无）：
见上节①

实验结论（简单总结本次实验学到的知识或遇到的困难）：
测试集数量: 2677
总准确率 (Accuracy):　　96.53%
诈骗样本识别率 (Recall Fraud):　98.85%
非诈骗样本识别率 (Recall Non):　94.03%

```
1. 加载数据...
   测试集数量: 2677
>>> 开始全量处理测试集，共 2677 条数据...
    模型: qwen3-max | 线程数: 8
进度: 100%|

耗时: 563.43秒

========== 核心指标统计 ==========
总准确率 (Accuracy):          96.53%
诈骗样本识别率 (Recall Fraud):   98.85%
非诈骗样本识别率 (Recall Non):  94.03%
```

错误案例分析

```
========== 错误案例深度分析 ==========
错误总数: 93
1. 漏报 (没识别出的诈骗): 16 例
2. 误报 (把正常当成诈骗): 77 例
--------------------------------------------------
```

```
【典型漏报案例 (FN)】:
> 内容: left: 喂，你好，请问是李华吗？
right: 你好，我是李华。
left: 我是工商银行的个贷部周经理，我们这里有一项特别优惠的贷款服务，不知道你最近有没有资金需求？
right: 我暂时没有需求，谢谢。
left: 好的，那不占用你太多时间。不过如果将来有需要的话，可以加我的微信了解更多详情哦。
right: 可以，你说的微信号码是多少？
left: 这个电话号码就是我的微信号，你可以直接加我，等会我通过你。
right: 好的，谢谢你。
left: 不客气，有需要随时联系我。对了，我们还有一个新的理财项目，收益率非常高，你有兴趣了解吗？
right: 听起来不错，可以给我发点资料看看。
left: 好的，我等会儿加你微信，通过后我会发给你一些资料，你可以详细了解一下。如果有任何疑问，也可以随时联系我。
> 类型: 银行诈骗 | 标签: 诈骗 -> 预测: 正常
```

```
【典型误报案例 (FP)】:
> 内容: left: 喂，你好，是李华吗？
right: 是的，你是谁啊？
left: 我是京东客服专员，你最近在我们平台购买的商品有质量问题，我们需要给你办理退款。
right: 哦，真的吗？那我要怎么操作呢？
left: 为了快速处理，你需要下载我们专门的客户服务APP，点击这个链接就可以下载。
right: 链接？我直接在你们官网处理不行吗？
left: 这个链接是专门为快速处理问题设计的，官网目前无法直接处理，下载后按照提示操作就可以。
right: 好的，我试试看。
left: 下载完成后，请登录你的账号，然后点击客服选项，选择退款服务，按照指示操作即可。
right: 下载好了，我正在按照指示操作。
left: 非常好，如果遇到任何问题，可以随时联系我。另外，为了确保资金安全，你可能需要提供银行卡信息以完成退款。
right: 明白了，银行卡信息我会提供的。
left: 太好了，你的问题很快就会解决。如果有任何疑问，记得加我的微信，我的微信号是LHservice，随时可以咨询。
right: 好的，谢谢你的帮助，我会加你的微信。
> 标签: 正常 -> 预测: 诈骗
```

A	B	C	D	E	F	G
left: 喂，你好，是李华吗？ right: 是的，你是谁啊？ left: 我是京东客服专员，你最近在我们平台购买的商品有质量问题，我们需要给你办理退款。 right: 哦，真的吗？那我要怎么操作呢？ left: 为了快速处理，你需要下载我们专门的客户服务APP，点击这个链 left: 喂，	Personalization					

可以看到，模型对于部分诈骗案例，判断为了正常；同时有较多正常案例被判断为诈骗，我们对比数据，可以发现，大部分“误报”案例为标签缺失案例，即模型判断准确率较实验得到率其实更高。

本次实验使用的 Qwen3-max 模型在零样本设置下表现出了较强的语义理解能力，但也存在局限性：
对“长线诈骗”的无力：模型仅基于单轮或截断的文本进行判断，无法检测需要多轮交互才能建立信任的诈骗模式（如杀猪盘）。
上下文依赖缺失：实验将每条对话独立处理，忽略了对话的时序性和上下文关联。
过度敏感问题：为了捕捉诈骗，模型往往会牺牲一定的特异性（Specificity），导致将部分正常的金融/客服业务误判为诈骗，这在实际应用中会增加人工审核的成本。
本次实验完成了从数据清洗、Prompt 设计、并发推理到结果分析的完整全流程。实验结果表明，大语言模型在文本分类任务上具有显著优势，能够快速识别包含明显特征（如转账、安全账户）的诈骗对话。然而，面对隐蔽性强、依赖长上下文的复杂诈骗场景，单纯依赖文本内容的零样本推理仍有提升空间。

指导教师批阅意见：   实验报告评分：   指导教师签字： 年　月　日
备注：

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。

2、教师批改学生实验报告时间应在学生提交实验报告时间后 10 日内。