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 2：对话行为与策略建模

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 26 日 - 2026 年 01 月 04 日

实验报告提交时间：2026 年 01 月 04 日

教务部制

实验目的与要求：

（1）掌握对话行为与交互策略（interaction_strategy）建模的基本概念，理解两者在对话语义分析中的互补关系。

（2）通过大语言模型自动生成对话中，每句话的行为类别标签（如请求、陈述、确认、拒绝等），实现从零标注的语义行为识别。

（3）在获得的对话行为类别基础上，使用统计或者其它分析方法，分析是否存在以下关系。例如但不限于以下关系：

- “请求 + 低真实性”常出现在“诈骗”对话中；
- “陈述 + 个性化”更常出现在“真实”对话中；
- 分析联合建模结果在提升诈骗检测效果方面的潜力。

方法、步骤：

（一）数据准备

1. 使用与实验 1 相同的中文对话数据集，字段包括：
   * `specific_dialogue_content`：对话文本；
   * `interaction_strategy`：交互策略标签（Clarity、Personalization、Relevance、Completeness、Truthfulness 等）；
   * `is_fraud`：是否诈骗；
   *说话者角色（如客服、用户、诈骗者）。
2. 将对话按轮次划分，提取每一轮对话中的 left 端的所有句子（即主动方或主要说话方的发言），作为对话行为分析的基本单元。

（二）对话行为类别自动识别

任务定义： 利用大语言模型（如 Qwen、ChatGLM3、LLaMA 或 其它模型）为每个 left 端句子生成其对应的对话行为类别。

对话行为类别示例：

* 请求类（Request）：询问、索取信息、提出要求；
* 陈述类（Statement）：提供信息、描述事实；
* 确认类（Confirmation）：肯定、确认对方观点；
* 拒绝类（Rejection）：否定、拒绝请求或提议；
* 其他类（Other）：不属于以上类别的发言。

方法与实现（仅供参考，可自行定义）：

1. Prompt 设计：
   设计引导式提示词，引导模型理解任务语义。
   示例：
   > “请判断以下句子的对话行为类别（请求、陈述、确认、拒绝、其他）并输出类别名称：[句子内容]”
2. 模型执行：
   * 选用大模型；
   * 使用 Python 或 API 或对话窗口，调用接口批量处理所有 left 端句子；
   * 将输出结果保存为新字段 `speech_category`：计数向量：[#请求, #陈述, #确认, #拒绝, #其他]；

3. 结果验证：

* 随机抽取部分结果进行人工校对；
* 统计一致率（模型预测与人工判断的一致比例）；
* 若一致率 ＜0.8，可微调 prompt 或重新采样模型输出。

（三）对话交互策略`interaction_strategy`、对话行为类别 `speech_category`、对话诈骗标签`is_fraud`的联合分析

（必做题）任务一：分层共现统计（行为 × 策略 × 诈骗）

目标：找出在诈骗/非诈骗里更常见的“对话行为类别 × 交互策略”组合。

步骤（仅供参考，可自行定义）：先把每个对话的 left 端句子汇总成行为占比，再与该对话的 interaction_strategy 配对，分别在 `is_fraud=1/0` 两组计算每个“行为×策略”的占比与提升率并排序。

例子：诈骗组“请求 × 低真实性”占比 18%，非诈骗组 5%，提升率=18/5=3.6 ⇒ 更偏向诈骗；非诈骗组“陈述 × 个性化”占比 22%，诈骗组 7%，提升率=22/7=3.14 ⇒ 更偏向真实。

（选做题）任务二：轻量解释建模（量化组合对诈骗概率的影响）

目标：量化不同“行为×策略”组合对 `is_fraud` 的指示力度，形成可操作结论。

参考步骤：用对话级特征（行为占比、策略 one-hot）训练一个逻辑回归或浅层决策树预测 `is_fraud`，直接用特征重要性判定哪些组合最关键。

例子：模型系数显示“请求 × 低真实性”为正且数值较大 ⇒ 诈骗概率上升；“陈述 × 个性化”为负 ⇒ 更可能是非诈骗对话。

实验过程及内容：

（挑选关键部分，截图运行结果）

任务一：分层共现统计（行为 × 策略 × 诈骗）

数据准备：使用原始 CSV 文件 测试集结果.csv，字段包含 specific_dialogue_content、interaction_strategy、is_fraud 等；每行为一段对话文本。

句子单元构建: 将每条对话按“轮次”划分，提取所有 left: 标记的发言作为基本单元（每条发言为一句或一个片段），形成每条对话的 left 端句子列表。实现文件：run_experiment_llm.py 中函数 extract_left_utterances()。

```python
def extract_left_utterances(dialogue_text):
    # 提取所有 'left:' 后面的发言块，作为一个 utterance 单位
    # 适配多行字段
    parts = re.split(r'(?i)left:\s*', dialogue_text)
    utterances = []
    for p in parts[1:]:
        # 截取到下一个 right: 或 end
        u = re.split(r'(?i)right:\s*', p)[0].strip()
        # 规范化空白
        u = re.sub(r'\s+', ' ', u)
        if u:
            utterances.append(u)
    return utterances
```

对话行为自动识别：设计单句 prompt（示例见脚本 build_prompt()），目标五类：请求、陈述、确认、拒绝、其他。脚本支持两种模式：

```python
def build_prompt(utterance):
    return (
        "请判断下列句子的对话行为类别，仅在五类中选择并只输出类别名称（不要多余说明）：\n"
        "类别选项：请求、陈述、确认、拒绝、其他。\n"
        f"句子：{utterance}\n"
        "请只返回其中之一：请求 或 陈述 或 确认 或 拒绝 或 其他。"
    )
```

LLM 模式：调用兼容 OpenAI 接口的模型批量预测（可通过 Config.USE_LLM=True 启用）；

规则回退：若不能调用外部 API，使用内置简易规则快速分类（当前实验为加速使用规则回退）。实现：classify_utterances()、call_llm()、rule_based_classify()。

批量执行与去重加速：将所有对话的 left 句扁平化去重后批量请求/分类，结果再回填到每个对话，避免重复调用同一句子的标注。

```python
def call_llm(prompt, config: Config, max_retries=3):
    # 使用 OpenAI 兼容接口（若 openai 可用），否则返回 None
    if openai is None:
        return None
    openai.api_key = config.API_KEY
    openai.api_base = config.BASE_URL
    for attempt in range(max_retries):
        try:
            resp = openai.ChatCompletion.create(
                model=config.MODEL_NAME,
                messages=[
                    {"role": "system", "content": "你是一个中文对话发言行为分类器，回答必须严格为指定的类别中文名称。"},
                    {"role": "user", "content": prompt},
                ],
                temperature=0.0,
                max_tokens=32,
            )
            text = resp['choices'][0]['message']['content'].strip()
            # 只保留首个词
            text = text.splitlines()[0].strip()
            # 规范化
            for k in CATEGORY_MAP:
                if k in text:
                    return CATEGORY_MAP[k]
            # 如果返回英文或其他，尝试映射
            if text.lower().startswith('request'):
                return '请求'
            if text.lower().startswith('statement'):
                return '陈述'
            if text.lower().startswith('confirmation'):
                return '确认'
            if text.lower().startswith('rejection'):
                return '拒绝'
            return '其他'
        except Exception as e:
            wait = 1 + attempt * 2
            time.sleep(wait)
    return None
```

结果保存: 为每条对话生成行为计数与占比列并保存完整预测文件：测试集结果_LLM 全量预测.csv。针对每条对话计算每类行为在 left 句中的占比（行为占比特征），按 interaction_strategy 分组，在 is_fraud=1/0 两组分别计算每个“行为×策略”的平均占比与提升率（uplift = mean_fraud / mean_nonfraud），并按提升率排序输出为 测试集_行为策略共现分析.csv。

任务二：轻量解释建模（量化组合对诈骗概率的影响）

特征构造: 每条对话的行为占比特征（prop_请求/陈述/确认/拒绝/其他） + interaction_strategy one-hot + 行为×策略交互项（如 prop_请求_x_strat_Clarity）。

模型与训练: 采用带 class_weight='balanced' 的 LogisticRegression（liblinear），按 80/20 做分层拆分（stratify），训练并在测试集上计算 AUC/accuracy/precision/recall。

```python
def train_and_explain(X, y, feature_names):
    # stratified split
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42, stratify=y)

    clf = LogisticRegression(max_iter=2000, class_weight='balanced', solver='liblinear')
    clf.fit(X_train, y_train)

    y_prob = clf.predict_proba(X_test)[:, 1]
    y_pred = clf.predict(X_test)

    metrics = {
        'roc_auc': roc_auc_score(y_test, y_prob),
        'accuracy': accuracy_score(y_test, y_pred),
        'precision': precision_score(y_test, y_pred, zero_division=0),
        'recall': recall_score(y_test, y_pred, zero_division=0),
        'n_train_pos': int(y_train.sum()),
        'n_train_neg': int((1 - y_train).sum()),
    }

    coefs = clf.coef_[0]
    coef_df = pd.DataFrame({'feature': feature_names, 'coef': coefs})
    coef_df['abs_coef'] = coef_df['coef'].abs()
    coef_df = coef_df.sort_values(by='coef', ascending=False)
    return clf, metrics, coef_df
```

feature	coef	abs_coef
prop_请求	1.846378	1.846378
prop_陈述	1.509251	1.509251
strat_Relev	1.460979	1.460979
prop_请求	1.403509	1.403509
prop_请求	1.175177	1.175177
prop_陈述	1.052981	1.052981
strat_Persc	0.920123	0.920123
strat_Clarit	0.749564	0.749564
prop_确认	0.610645	0.610645
prop_确认	0.521961	0.521961
prop_确认	0.389909	0.389909
prop_拒绝	0.289182	0.289182
prop_请求	0.273797	0.273797
prop_确认	0.127823	0.127823
prop_拒绝	0.006377	0.006377
prop_其他	0	0
prop_其他	0	0
prop_其他	0	0
prop_拒绝	0	0
prop_其他	0	0
prop_其他	0	0
prop_其他	0	0
prop_陈述	-0.10182	0.101821
prop_拒绝	-0.12735	0.127352
prop_确认	-0.20261	0.20261
prop_确认	-0.22644	0.226437
prop_请求	-0.25917	0.259166
prop_拒绝	-0.62171	0.621707
prop_请求	-0.74694	0.746939
prop_拒绝	-0.78991	0.789914
strat_Com	-1.0752	1.075197
prop_陈述	-1.23676	1.236756
prop_陈述	-1.39278	1.392776
strat_Factu	-1.61293	1.612928
prop_陈述	-2.61643	2.616431

正向（提示诈骗）的 top5:

prop_请求: 1.846

prop_陈述_x_strat_Personalization: 1.509

strat_Relevance: 1.461

prop_请求_x_strat_Factual Authenticity: 1.404

prop_请求_x_strat_Clarity: 1.175

负向（提示非诈骗）的 top5:

prop_陈述_x_strat_Factual Authenticity: -2.616

strat_Factual Authenticity: -1.613

prop_陈述: -1.393

prop_陈述_x_strat_Clarity: -1.237

strat_Completeness: -1.075

数据处理分析（如不涉及可写无）：
见实验过程

实验结论（简单总结本次实验学到的知识或遇到的困难）：

任务一：分层共现统计（行为 × 策略 × 诈骗）

strategy	behavior	mean_pro	mean_pro	uplift	n_fraud	n_nonfrauc
Relevance	拒绝	0.002222	0	inf	75	4
Clarity	拒绝	0.008695	0.002632	3.30395	346	76
Clarity	请求	0.540834	0.206407	2.620236	346	76
Factual Au	请求	0.645741	0.249118	2.592104	570	1173
Relevance	请求	0.622175	0.4	1.555437	75	4
Personaliz	请求	0.525535	0.388103	1.354112	396	35
Clarity	确认	0.194199	0.146664	1.324112	346	76
Factual Au	确认	0.209421	0.205295	1.0201	570	1173
Personaliz	确认	0.164187	0.192457	0.853108	396	35
Personaliz	陈述	0.289247	0.347759	0.831744	396	35
Relevance	确认	0.137386	0.2	0.686931	75	4
Relevance	陈述	0.238217	0.4	0.595542	75	4
Factual Au	拒绝	0.005029	0.009852	0.510499	570	1173
Clarity	陈述	0.256272	0.644298	0.397754	346	76
Personaliz	拒绝	0.021032	0.071681	0.293409	396	35
Factual Au	陈述	0.139809	0.535735	0.260966	570	1173
Relevance	其他	0	0	0	75	4
Personaliz	其他	0	0	0	396	35
Clarity	其他	0	0	0	346	76
Factual Au	其他	0	0	0	570	1173
Completer	请求	0	0.7	0	0	2
Completer	陈述	0	0.1	0	0	2
Completer	确认	0	0.2	0	0	2
Completer	拒绝	0	0	0	0	2
Completer	其他	0	0	0	0	2

在样本量较充足且提升率显著的组合中，Clarity + 请求（诈骗组平均占比 0.5408，非诈骗 0.2064，提升率 ≈ 2.62，n_fraud=346, n_nonfraud=76）和 Factual Authenticity + 请求（诈骗 0.6457，非诈骗 0.2491，提升率 ≈ 2.59，n_fraud=570, n_nonfraud=1173）对诈骗更具区分力，说明“请求类发言在低真实性/事实类策略下更常见于诈骗对话”。

拒绝类信号: Clarity + 拒绝 提升率 ≈ 3.30（n_fraud=346, n_nonfraud=76），同样提示在部分策略下拒绝类也偏向诈骗组，但绝对占比仍较低（均值 < 0.01）。

中性/偏向非诈骗: 若干组合（如 Clarity + 陈述、Factual Authenticity + 陈述）在非诈骗组占比更高（uplift < 1），表明“陈述 + 个性化/真实性良好”更常见于真

实对话的直观结论在数据中部分成立。

任务二：轻量解释建模（量化组合对诈骗概率的影响）

roc_auc	accuracy	precision	recall	n_train_po	n_train_neg
0.902662	0.837687	0.849817	0.834532	1109	1032

“请求”类占比是最强的正向指示因子，且在若干策略（特别是低真实性/Clarity/Factual Authenticity）下的交互项进一步增强了诈骗信号，支持实验假设“请求 + 低真实性/特定策略更常出现在诈骗对话”。

与之相对，“陈述”及与真实性相关的策略倾向于降低诈骗概率（表明真实对话更偏陈述/真实性较高）。

指导教师批阅意见：

实验报告评分：

指导教师签字：

2026 年 01 月 04 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。