模式识别技术方案与实现-v1

1. 方案

按照之前的方案，backend包含5个主要的模块：**预处理、模式识别、日志总结、意图预测、能力分析**。

上周实现了**预处理模块**，主要完成如下工作：

1. **event_type转换**：前端收集的事件类型基于API，能进一步总结
2. **剔除冗余数据**：看数据集情况
3. **形成历史操作件artifact_history**：用于后续的预测检索
4. **形成历史命令cmd_history**：用于后续的预测检索

本周继续实现**模式识别模块**。

1.1. 目标与思路

模式识别模块需要识别一个行为所处的状态[配置环境/调查阅读/编写内容/执行验证/其它]。

考虑到(1) 完全基于规则来分类复杂度比预想的大；（2）后期可能会细化模式种类，改一次就要刷一遍规则。

因此，目前计划先用cfc做分类任务，先用一个能力强的模型看看效果。

1.2. 特征工程

总特征维度：287维，包含三大类特征：**事件类型特征、制品特征 和 上下文特征**。

特征选择需要尽可能捕捉到多维度的语义信息，同时**考虑根据人的经验制定规则，对部分特征进行增强**，以强化模型的性能。

1.2.1 事件类型特征 (20维)

- 使用 one-hot 编码表示不同事件类型

1.2.2 制品特征 (146维)

包含 基础特征 + 语义增强特征 + Word2Vec词嵌入特征

（1）基础特征

- a. 制品类型的 one-hot 编码
- b. 名称长度、层级深度等数值特征

（2） 语义增强特征

- a. 路径特征
  - Case1：识别目录名称。根据鸿蒙工程的规范，如果操作工件所处目录中包含"main/pages"，说明大概率正在执行编写内容；包含"mock"、"test"等，说明大概率执行验证工作。
  - Case2: 识别文件名称及后缀。例如操作`.md`文件，说明可能在调查阅读。
  - Case3: 识别具体工件的名称。例如"xxxView"、"xxxController"、"xxxTestCase"等。
- b. 命名规范特征
  - 例如全大写、全小写、下划线命名、驼峰命名等。
- c. 引用特征
  - 引用存在性
  - 引用数量

（3）Word2Vec词嵌入特征

- 使用 Word2Vec 模型生成词嵌入向量，处理文件路径和名称的语义信息，捕捉命名模式的相似性。

1.2.3 上下文特征 (121维)

包含 基础特征 + 语义增强特征 + Word2Vec词嵌入特征

（1）基础特征

- 上下文类型的one-hot编码
- 内容长度和位置特征
  - 修改前后内容长度
  - 内容变化量
  - 起始/结束行号和字符位置
  - 修改范围（跨越行数）

（2） 语义增强特征

- Case: 对于终端命名，识别是否是npm命令、python命令、git命令、管道命令、重定向等。

（3）Word2Vec词嵌入特征

- 使用 Word2Vec 模型生成词嵌入向量，捕捉上下文信息的相似性。

引入Word2Vec词嵌入会导致维度大规模上升。如果可以完善更多的语义增强特征，确保保留足够丰富的语义特征，可以删除词嵌入特征。需要试验。

2. 实现

2.1. 代码组织

- dataset.py: 数据处理和特征工程
- learner.py: 模型训练和验证逻辑
- config.py: 配置参数管理

2.2. 进度

跑通了数据预处理+特征工程+训练流程。

```
Found 10 pt files in dataset
Processing 2024-12-09 19 copy 8.pt...
Processing 2024-12-09 19 copy 5.pt...
Processing 2024-12-09 19 copy.pt...
Processing 2024-12-09 19 copy 7.pt...
Processing 2024-12-09 19 copy 2.pt...
Processing 2024-12-09 19.pt...
Processing 2024-12-09 19 copy 6.pt...
Processing 2024-12-09 19 copy 3.pt...
Processing 2024-12-09 19 copy 9.pt...
Processing 2024-12-09 19 copy 4.pt...
Dataset X shape: torch.Size([10, 67, 185])
Dataset Y shape: torch.Size([10, 67])
```

```
Epoch 16: 100%|
████████████████████████████████████████████████████████████████████████
██████████████████████████| 1/1 [00:00<00:00,  2.64it/s,
v_num=1]Metric val_acc improved by 0.015 >= min_delta = 0.0. New best
score: 0.970

Epoch 24: 100%|
████████████████████████████████████████████████████████████████████████
██████████████████████████| 1/1 [00:00<00:00,  2.63it/s,
v_num=1]Metric val_acc improved by 0.015 >= min_delta = 0.0. New best
score: 0.985

Epoch 33: 100%|
████████████████████████████████████████████████████████████████████████
██████████████████████████| 1/1 [00:00<00:00,  2.64it/s,
v_num=1]Metric val_acc improved by 0.015 >= min_delta = 0.0. New best
score: 1.000

Epoch 53: 100%|
████████████████████████████████████████████████████████████████████████
██████████████████████████| 1/1 [00:00<00:00,  2.63it/s,
v_num=1]Monitored metric val_acc did not improve in the last 20
records. Best score: 1.000. Signaling Trainer to stop.

Epoch 53: 100%|
████████████████████████████████████████████████████████████████████████
██████████████████████████| 1/1 [00:00<00:00,  2.62it/s, v_num=1]
Best validation accuracy: 1.0000
(VirtualMe) suyunhe25@dell-gpu-02:~/virtualme-backend$
```