## 页面分类辅助大语言模型引导策略方法报告

1. **研究背景**：

在移动应用自动化UI测试中,存在UI陷阱问题。即测试时应用长时间停留在某一（组）特定的页面中。陷入UI陷阱会导致测试效率降低，测试覆盖场景丢失等问题。目前对于UI陷阱的处理主要有规避进入UI陷阱和引导脱离UI陷阱两种方式。引导脱离UI陷阱需要设计引导策略或者人工编写引导脚本。随着大语言模型逻辑推理能力的不断提高和知识库的不断丰富，我们希望利用大语言模型引导测试工具脱离UI陷阱。

二. **主要问题**：

1. UI陷阱识别问题：

如何判断测试工具是否陷入UI陷阱？

2.UI语义信息问题：

怎么把当前的UI转化为大语言模型能够接受的文本化语义信息？

3.准确率问题：

怎么提高大语言模型识别出正确的脱离方式的概率。

4.表单问题：

怎么通过需要填写符合特定规则的文本的表单。

5.路径多样化度量问题

怎么度量探索路径组合的多样性

**三． 主要工作：**

A. UI陷阱识别问题：

尝试过的方法有：

1. 停留在某一activity的连续事件数量达到一定阈值：

缺点：应用的某一activity可以具有多种业务功能，以activity作为划分粒度过大。

1. 生成当前UI的文本描述，若文本描述相同的连续事件数量达到一定阈值：

技术：生成当前UI文本描述的技术在后文UI语义信息问题中说明。

缺点：UI组件的内容数据（例如文本框中的文本）变动也会导致UI文本描述变动，这种方法的粒度过小。

1. 前后UI截图相似的连续事件数量达到一定阈值：

技术：通过前后图片的dhash值计算相似度，相似度高于一定阈值认为相似。

优点：粒度适中。内容数据变动一般仍然是相似图片，业务功能变动一般不再是相似图片。

缺点：依赖于图片相似度，存在误判问题。

最后采纳了前后UI截图相似的连续事件数量达到一定阈值的方法。

B: UI语义信息问题：

方法：

1.通过组件的enabled，visible，covered等属性找到可交互组件。

2.通过clickable, editable, scrollable, checkable, long\_clickable等属性区分交互方式。

3.通过text，resource\_id, description等属性得到一些组件描述。

4.生成当前UI的文本描述，一个组件描述对应一个事件，询问大语言模型，根据回答执行事件。

优点：如果开发时text，resource\_id, description具备良好的描述信息，则可以提取到良好的UI语义信息。

例子：

一个真实输出的节选：

- a clickable view with id "fab\_expand\_menu\_button"(0);

- a clickable view with id "fab\_note"(1);

- a clickable view with id "fab\_checklist"(2);

- a clickable view with id "fab\_camera"(3);

- a clickable view with id "root"(4);

- a clickable view with id "root"(5);

- a clickable view with description "More options"(6);

- a clickable view with description "Sort" with id "menu\_sort"(7);

- a clickable view with description "Search" with id "menu\_search"(8);

可以看到排序（sort），搜索（search），更多选项（more options）等等语义清楚的组件信息。

缺点：如果开发时没有良好的描述，或者通过框架生成了没有描述的组件，则无法提取出良好的组件信息。

C.准确率问题：

方法：基于视觉分类的界面语义提取

简介：先通过截图识别当前页面的分类，根据分类启发式的调整提示词优化llm的表现。

技术：OCR文字识别，做出文字区域非文字区域图，对图做聚类训练，根据所得模型对页面分类。

详细介绍：

1.页面分类：

尽管LLM已经能分别获得UI界面中所有组件的详细信息，但页面整体的特征信息仍然有所欠缺。Safwat Ali Khan等人总结了UI界面的21种设计模式，高度凝练了UI界面包含的结构信息，一定程度揭示了界面的语义。通过识别页面的类型，并将类型信息反馈给LLM， 能够让LLM更好地理解界面的语义信息，从而在理解界面功能的基础上给出探索更多页面的指导。详细分类信息如下：

label\_mapping={0: "Advertisement", 1: "Calendar and time weather", 2: "Catalog ", 3 : "Feed ", 4: "Form ", 5:"Home menu ", 6:"List", 7:"Log-in ", 8:"Map ",9: "Onboarding ", 10: "Player ", 11: "Pop up menu", 12: "Product ", 13: "Search ", 14: "Settings ", 15: "Splash ", 16: "Terms and conditions", 17: "Travel booking", 18: "Type message", 19: "Viewer", 20: "Web browser"}

2.分类技术：

使用多模态的深度学习模型，结合视觉和文本来识别屏幕类型。该模型通过分析UI屏幕的截图，UI结构信息展示图以及OCR界面纯文本信息三个方面，将屏幕较准确地分类为预定义的设计模式。在prompt中加入页面类型信息能有效提高LLM对UI界面的理解。

3启发式方法：

在所有21种UI设计模式中，由于每种UI界面设计模式都沿用了相似的设计，可以认为，同一类UI界面能使用相似的方法跳出UI陷阱，进入下一个页面。对于大部分，特别是不需要文本输入的UI界面，可以通过简单的触摸或滑动等操作脱离UI陷阱。通过总结Aurora框架中的启发式方法以及软件测试过程中的实践，根据典型UI界面的特征，我们给出了路径引导策略的启发式方法，用于指导LLM快速选择到目标组件。实际使用的启发式方法如下：

action\_mapping={ 0:"clickable view with 'close' or some similar words and go back key.", 1:"clickable view.", 2:"clickable view.", 3:"clickable view with 'like' or 'comment' or some similar words and Scroll. ", 4:"clickable view with 'create' or 'ok' or 'done' or some similar words . ", 5:"clickable view.", 6:"clickable view with 'item' or some similar words", 7:"clickable view with 'create' or 'ok' or 'done' or some similar words.", 8:"clickable view.", 9:"clickable view with 'Next' or 'Skip' or 'Close' or some similar words.", 10:"clickable view with 'play' or some similar words.", 11:"clickable view with 'close' or some similar words and go back key.", 12:"clickable view with 'image' or 'product' or some similar words.", 13:"clickable view .", 14: "clickable view.", 15: "clickable view.", 16:"clickable view with 'Skip' or 'accept' or some similar words.", 17: "clickable view.", 18:"clickable view with 'send' or 'ok' or 'done' or some similar words.", 19:"clickable view with 'video' or 'image' or some similar words.", 20:"clickable view with 'close' or some similar words and go back key." }

4．提示词总览

（1）任务说明：You are an expert in App GUI testing. Currently, the App is stuck on the MainActivity page, unable to explore more features. Your task is to select an action based on the current GUI Information to perform next and help the app escape this page.

（2）页面类别及通常方案说明

" This is most likely a(an) "

+self.label\_mapping[page\_classify\_idx].format()

+f" page."

+f" Please give priority to "

+self.action\_mapping[page\_classify\_idx].format()

其中self.label\_mapping 存储了页面的识别结果

self.action\_mapping 存储了通常的操作方案。

（3）页面状态与组件信息，以一个真实输出的节选为例：

- a clickable view with id "fab\_expand\_menu\_button"(0);

- a clickable view with id "fab\_note"(1);

- a clickable view with id "fab\_checklist"(2);

- a clickable view with id "fab\_camera"(3);

- a clickable view with id "root"(4);

- a clickable view with id "root"(5);

- a clickable view with description "More options"(6);

- a clickable view with description "Sort" with id "menu\_sort"(7);

- a clickable view with description "Search" with id "menu\_search"(8);

（4）历史操作信息：

I have tried following actions you gave me to leave this page but failed：- enter "I accept the terms and conditions." into view with text "Legal Notice: This application is for in..." with id "detail\_content";

- enter "I agree to the terms and conditions." into view with text "-J" with id "detail\_title"

记录了在这个UI陷阱中已经进行过的历史操作，辅助大模型选择。

（5）输出规约：

Which action should I choose next? Just return the action id and nothing else.

便于处理大模型给出的结果。

优点：对于登录（onboarding）弹窗（pop-up）广告(advertisement) 等页面表现优异，通常只需一次询问即可得到正确的操作。

缺点：依赖于识别准确率，无法准确识别则无法发挥效果。

D. 表单问题:

方法简介：在页面存在可编辑文本框时，先编辑可编辑文本框。

介绍：

（1）方法说明

在大部分页面模式中，可编辑组件的出现可能都较低。但在少数页面模式，如搜索页 面或表格页面等，可编辑组件是页面的关键组件。为解决这些特殊页面的UI陷阱问题，本方法在页面中存在未被LLM填写过的文本框时直接触发文本框提示词，直接让LLM根据页 面语义填写文本框，以支持后续的探索操作。

（2）提示词信息

任务说明和组件信息不变

（3）补充任务说明和输出规约：

The application requires legal input to continue exploring. What text

should I edit to view with text "-J" with id "detail\_title"? Please give me a text you think rational and meaningful. Just return the text and nothing else.

优点：先填文本提高了llm通过表单以及搜索框等页面的概率。

缺点：无法保证llm可以填出符合业务场景规则的文本。

E.路径多样化度量问题：

路径多样性应当度量测试过程中探索路径的多样程度，路径多样性更高则更可能探索到一些“意想不到”的执行路径，并找到一些“难以发现”的缺陷。对于UI测试，我们可以将节点定义为某一特定结构的页面，将边定义为这些节点间的迁移，一条路径则是一个执行过程中可能出现的迁移边的序列。因此，要实现一个成熟的路径多样性度量策略，应当良好地定义怎么区分不同的特定页面，识别出页面间所有可能的迁移边，构造出类似于CFG图的迁移图，再按照基路径覆盖的方法计算一次执行的路径覆盖率。

我们利用了driodbot工具输出的utg图：

DriodBot在运行过程中根据识别的页面信息区分出不同的页面，记录运行中找到的这些页面间的跳转关系，并根据这些信息构造utg图，示例如下：

其中的节点是页面的截图示例，边是按程序运行出现顺序排序的跳转关系。图中的默认分类方式中，页面拥有相同的控件结构但具有不同的内容数据时会被分类为不同的页面，即两个页面仅仅是文本框内容不同也将成为不同的页面。这样的分类粒度过于精细，观察上图可以发现，图中的节点数较多，有大量的节点只和一到两个节点有跳转关系，例如图中的环路。

图形用户界面

AI 生成的内容可能不正确。

如果改变分类方式，拥有相同控件结构但内容数据不同时也分类为相同的页面，即按照结构聚集时，上图的utg将转变如下：

箱线图

AI 生成的内容可能不正确。

可以看到，总的状态数降低了，节点间的联系更紧密了，这样的结构更适合做基路径的计算

目前我们的策略是一个初步策略：

策略：通过统计结构聚集下的节点数和转移边数初步反应多样性。

理由：一个应用存在大量的页面和跳转关系，一次运行中探索出的节点数和转移边数与探索到的路径的多样性有一定的正相关关系。

缺点：没能真正实现路径级别的度量，节点数和边数跟路径多样性之间没有绝对的对应关系

**四． 实验评估**

1．原型实验评估

（1）评估标准

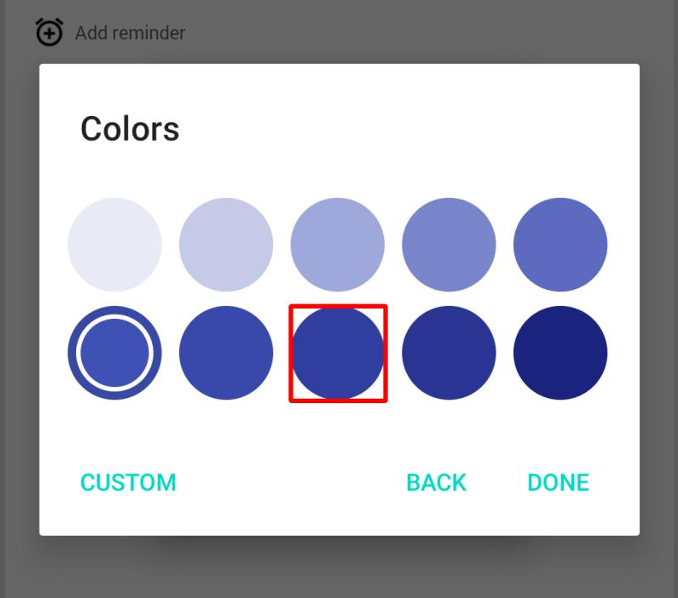
通过对移动应用页面特征的观察，我们认为ui陷阱大致分为两类。

A 类是通过交互按键可以脱离的ui陷阱，典型的有： ① 弹窗（广告，条款等，由于页面上的可交互组件多，难以关闭弹窗） ② 引导界面（需要特定交互序列）

B 类是需要输入文本后交互，典型的有： ① 表单 ② 消息输入页面 ③ 搜索框页面 它们需要在输入有意义的内容内容后与“提交”按钮交互。

下面将在我们的原型实验中选取两个具有代表性的例子解释我们的方法是如何分别解 决这两类UI陷阱的。

（2）第一类陷阱：



这是软件omninote中的颜色选择界面，

Kea给LLM的页面组件提示如下所示：

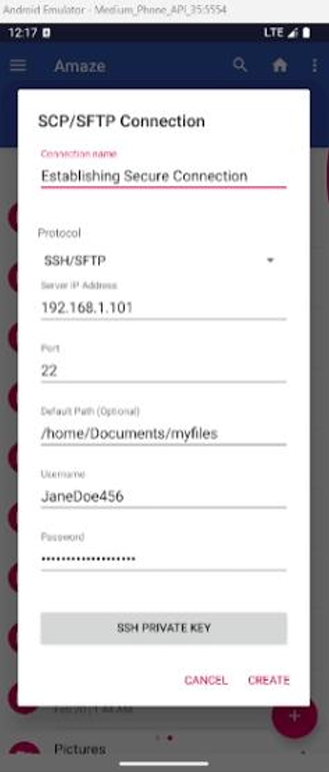
- a clickable view with text "DONE"(0); ...(1) ... ...(22) 其中(1)到(22)是选择颜色等无用组件，其中的BACK按钮只能返回到上一级颜色选择页面， 被认为不是离开UI陷阱的正确选择。

我们的策略生成的提示大致如下： Task This is most likely a(an) Pop up menu page. Please give priority to clickable view with 'close' or 'Done' or some similar words and go back key.

在这样的引导下一次查询模型便识别出Done按钮并交互。

评估：未经改善的LLM策略会多次选择其它的无效交互，random策略只有23分之1的概率选择期望的事件，因此这一策略对脱离ui陷阱有帮助。

（3）第二类陷阱



上图显示了amaze软件中的一个SCP/SFTP连接的表单。我们的策略是让大模型先给所有文本框构建输入，输入内容如图中所示（其中password为MyPassWord123）。之后通过识别结果生成提示词让LLM识别提交键，由于提示识别类似create，ok的按钮，LLM一次 查询便识别到create并交互。

评估：如果表单对于输入合法性要求不严格，这一策略可以通过表单，如果采用random策略，几乎不可能生成这样的通过序列，如果采用未改善的LLM策略，大模型也几乎不会按 照这一序列执行事件，因此这一策略对脱离ui陷阱有帮助。

2.覆盖率实验评估：

（1）使用android coverage tool测试

实验设置：

基线方法：kea 原random策略 100个事件重启

对照组1：kea图形相似识别+UI语义提取+先填文本的llm策略

对照组2：对照组1+页面分类的llm策略

实验数据集：应用trivago,omninotes,amaze,FoxNews,learn

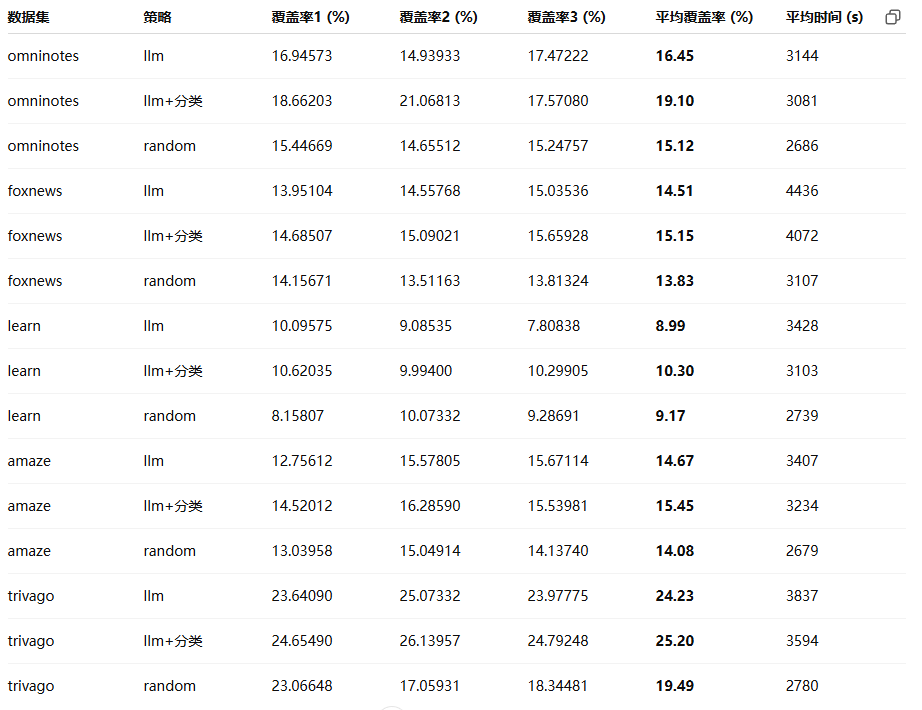
覆盖了多媒体密集型应用和逻辑业务密集型应用。

实验环境：个人笔记本 windows +安卓模拟器（android 15.0）

实验过程：测试500event后的覆盖率，每组每应用测试三次。

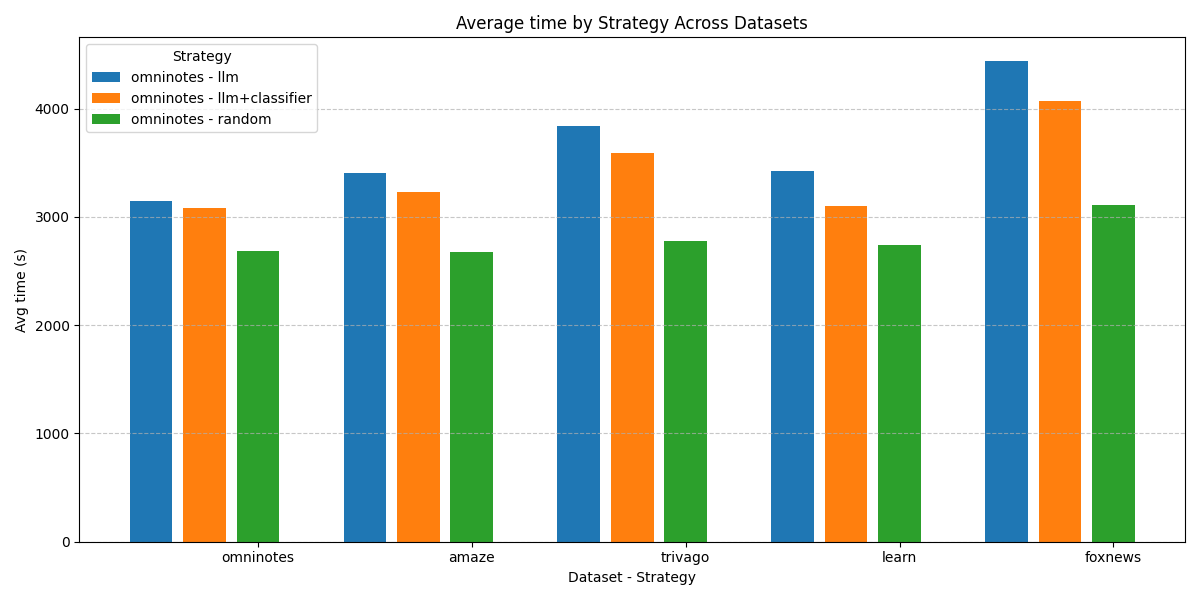
实验结果：

其中random表示基线，llm表示对照组1，llm+分类表示对照组2。



图表, 条形图

AI 生成的内容可能不正确。



（2）使用jacoco测试

实验设置：

基线方法：kea 原random策略 100个事件重启

kea 原random策略 100个事件无重启

对照组1：kea图形相似识别+UI语义提取+先填文本的llm策略

对照组2：对照组1+页面分类的llm策略

实验数据集：应用anki,myexpense,omninotes,vanilla

实验环境：个人笔记本 windows +安卓模拟器

实验过程：测试500event后的覆盖率，每组每应用测试五次。

实验结果：

Random无重启：kea 原random策略 100个事件无重启

Random有重启：kea 原random策略 100个事件重启

llm表示对照组1，llm+分类表示对照组2

**表格

AI 生成的内容可能不正确。**

结果分析：

1. 在acv组的测试数据中，llm+分类在所有应用中表现出了最高的平均覆盖水平。
2. 在jacoco组的测试数据中，llm+分类 在四个维度的覆盖率指标上，

在anki,omninotes,vanilla三个应用上均表现出最高水平，在myexpense应用上，三个维度略低于random有重启,一个维度略高于random有重启，认为总体上与random接近，并优于其余策略。

1. 总体上来看，在不考虑测试时间的情况下， llm+分类具有最好的表现。
2. 考虑时间的情况下，llm用时最长，llm+分类略短于llm,random用时显著更低。这是因为random的耗时较低，识别到UI陷阱进入llm引导后耗时变高，而llm+分类可以更快的脱离UI陷阱，因此用时略短于llm.
3. 据我们小组所知，在fastbot上实现的kea随机遍历的速度已经非常快，因此我们认为对比现在版本kea的随机策略，如果采用相同时间的方式做实验意义不大。我们采用了相同event的实验方法，如果未来分类和llm的效率和推理能力能够得到大幅优化，这一方案就有可能起到一些效果。
4. Ui陷阱越多，尤其是其中适合llm+分类的onboarding，pop-up，advertisement等类别的ui陷阱越多，llm+分类的优势越明显，这是不同应用之间涨幅具有较大差别的一个原因。另一原因是，覆盖率结果具有随机因素。
5. 从原型实验的效果评估，我们认为脱离ui陷阱的能力的确提高了，不过脱离ui陷阱能力提高在覆盖度上的反应并不明显。如果要提高覆盖度，应当做覆盖引导的进一步工作。脱离ui陷阱，在我们的测试过程中，本就是为了覆盖更多场景做出的努力之一。

**五. 总结**

方法总结：

在测试时识别图像相似度，当相似度大于一定阈值的连续事件数达到一定阈值后，对当前的截图做文本区域非文本区域图，交付给预训练的模型识别出页面的分类，对当前的UI进行文本化，把这一文本与该分类的提示交付与大模型，按照回答进行下一步事件，直到页面相似度低于阈值认为脱离出陷阱。

下一步工作：

1. 丰富每一分类的策略，增强健壮性。
2. 如何提高分类效率，如何提高llm交互效率，如何优化llm逻辑能力。
3. 完成路径级别的多样性度量。
4. 做覆盖引导的工作。

对自身工作评价：

优点：

1. llm+分类方法的尝试具有一定创新性。
2. 建立了充分的文档，详细地描述了工作内容。
3. 成品版本稳定，易于部署使用。
4. 使用了两个覆盖率工具测试，可以互为映衬。

不足：

1. 成品在覆盖率上不具备明显的优化效果。
2. 没有完成上述的进一步工作。

学到什么：

1. 安卓SDK工具链，安卓模拟器，安卓应用覆盖率测试的使用方法。
2. 安卓应用组件的存在形式，属性特点，信息获取方式。
3. 设计方法-原型实验-实现-大规模测试的方法论。
4. 设计提示-设计输出规约-软件对应操作的利用大模型辅助软件的方法。
5. Python项目的托管方式，多环境管理方式，部署方式，使用方式。
6. 团队协作与项目管理。