

BURGS Weekly Presentation

Broadening Undergraduate Research Groups

10/31/2025

Allie, Casie, Gayatri, Kim



Gaze XR: LLM Interpretations Overview

Completed (Summer work & Current)

01

App Policy Extractions

02

Google Gemini Response Extractions

03

Planned with Dr. JinYi Yoon for the project trajectory

In Progress

04

ROUGE Evaluation programming & sensor sorting algorithm

05

Research in other evaluations for LLM and how they work

06

Expand beyond Google Gemini to paid services (ChatGPT, ClaudeAI, etc.)

```
app_privacy_policy.py rouge_eval.py openai_eval.py sensor_eval.py gemini_prompting ▶ ...  
GenAI Prompt Engineering > rouge_eval.py > ...  
1  from rouge_score import rouge_scorer  
2  
3  # -----  
4  # Evaluates the generated response given (finds the path the txt file is in to open)  
5  # uses ROUGE evaluation scoring. There are 3 aspects that are evaluated shown below  
6  #  
7  # ROUGE-N: Precision through overlap of reference and response.  
8  # Quantifies overlap of N-grams (contiguous sequences of N items - typically words or char  
9  # between the system-generated summary and the reference summary. Provides sight on prec  
10 # the system's output by considering the matching N-gram sequences.  
11 #  
12 # ROUGE-L: Looks into COMMON synonyms to for accuracy (doesn't have to be word for word)  
13 # Calculates "Longest Common Subsequence" (LCS) between the system and reference summar  
14 # sequences of words (doesn't have to be contiguous) that appear in both summaries. Mor  
15 # similarity measure and helps capture shared information beyond strict word-for-word ma  
16 #  
17 # ROUGE-S: Paraphrasing flexibility wording measurement.  
18 # Skip-bigram (pair of words in a sentence that allows for gaps or words in between) foc  
19 # skip-bigram overlap between the system and reference, enabling the assessment of sente  
20 # similarity. Paraphrasing relationships between sentences and provide insights into the  
21 # information with flexible word ordering  
22 #  
23 # Reference to be used is the actual privacy policy. The generated responses (system) wi  
24 # -----  
25  
26 # initialize scorer, specifies the scores I want to use  
27 scorer = rouge_scorer.RougeScorer(['rouge1', 'rouge2', 'rougeL'], use_stemmer=True)  
28  
29
```

LLM Interpretation & Evaluation

- Prompt Engineering
 - [Prompt engineering guide](#) (google)
 - [Prompt engineering guide](#) (AI source?)
- Rouge
 - [Rouge score api](#)
 - [Calculate rouge score in python](#)
 - [Text summarization with transformers in python](#)
 - [Rouge Score Tutorial \(From Medium\)](#)
 - [Rouge score how to calculate](#)
- BLEU
 - [BLEU score evaluation class](#)
 - [BLEU score in python tutorial](#)
- GPT Evaluation
 - [Open AI evals](#)

Gaze XR: Consent Prompt Bypassing Overview

We're Here ✌

Timeline

- 01 Met with Paul
 - Data inference
 - Apply their project to ours

- 02 Manifest Files for checking out developer ignorance for eye tracking

- 03 Learn more about Paul's project, help them in any way, see if their project works without the prompt on our end

Gaze XR: Manifest Evaluations(Kim)

Plan

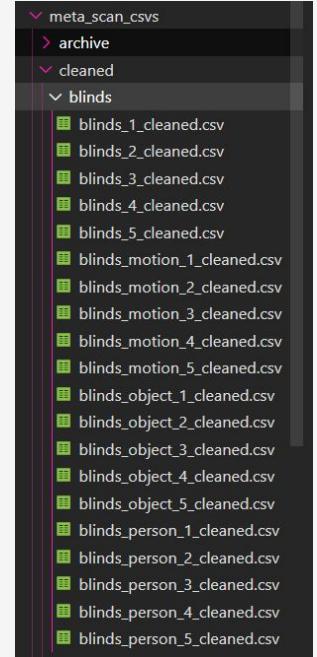
- Work with Paul and complete tasks as needed/directed
- Talked over Pauls work and determined we *should* be able to use the side channel attack with our original plan
- Need to confirm that the only thing that dictates the eye tracking tag is the declaration in the Android Manifests
- Work with Allie and Gaytri as they need extra hands

In Progress

- Created a Git Repo to contain the manifest evaluations
- Working on getting access to the manifests using the headset and desktop
- Compiling findings into a spreadsheet for easy comparison
 - manifest evaluations

Spatial Seer - Perfetto Data Scans

- We managed to collect 110+ scans and converted them to CSVs
 - 5 Room Types
 - Meeting room (blinds up)
 - Meeting room (blinds down)
 - Kitchen
 - Hallway
 - Lab room
 - 20+ Trials
 - 5 Basic scans
 - 5 Object Alteration
 - 5 White Noise (Person walking around scan)
 - 5 Motion (VR Headset User walking/looking around)
- MQ3 auto updated and it changed the way the headset scans, messed with the data we were collecting with Perfetto
 - Able to be fixed in post-processing but requires more/different cleaning techniques
 - Currently working on integrating all 100+ trials in ROCKET and SVM Models
 - Also turned off the auto update feature to protect us from this problem in future



Spatial Seer - Perfetto Data Scans

- The performance indicators that matter the most:
 - app_rss_mb → Virtual memory size in bytes
 - app_vss_mb → “Resident Set Size”, AKA the number of pages the process has in real memory
 - app_uss_mb → “Unique Set Size”, measures the amount of memory used exclusively by the process without considering shared memory

MANPAGES

Index About Manpages FAQ Service Information

debian / stretch / manpages / proc(5)

PROC(5) Linux Programmer's Manual PROC(5)

NAME
proc - process information pseudo-filesystem

DESCRIPTION
The *proc* filesystem is a pseudo-filesystem which provides an interface to kernel data structures. It is commonly mounted at */proc*. Most of it is read-only, but some files allow kernel variables to be changed.

Mount options
The *proc* filesystem supports the following mount options:

hidrepid=n (since Linux 3.3)
This option controls who can access the information in */proc/[pid]* directories. The argument, *n*, is one of the following values:

links
langu
packa
raw m
table
other
stretc
testin
stretc
unsta
other
Deuts
Englis

Spatial Seer - Unity Data Scans

- Trying to get the Unity Profiler Analyzer to work on our Unity scanning applications to reaffirm attacker's access to sensitive data
 - Hope to replicate work with Perfetto
 - However, no current way to export the data locally with Unity (as my research indicates)
- Found this script that supposedly can export the .data files and convert them into CSVs
- Integrating with both AR + MR Unity Applications for MQ3
- Gayatri and I are working on MR Unity Application next week

The screenshot shows a GitHub repository page for 'unity-profiler-data-exporter'. The repository is public and was forked from 'steve3003/unity-profiler-data-exporter'. The master branch has 2 branches and 0 tags. A message indicates that this branch is 2 commits ahead of the upstream master. The repository contains several folders: Allocators, Factories, StatsCalculator, UnityWrappers, and Utils. All code is described as being inactive outside the Unity Editor. The repository was last updated 6 months ago, with 40 commits. A file named 'ignore' is also present.

File/Folder	Description	Last Commit
Allocators	All code is inactive outside Unity Editor	6 months ago
Factories	All code is inactive outside Unity Editor	6 months ago
StatsCalculator	All code is inactive outside Unity Editor	6 months ago
UnityWrappers	All code is inactive outside Unity Editor	6 months ago
Utils	All code is inactive outside Unity Editor	6 months ago
ignore	ignore unity meta files	8 years ago

Spatial Seer - Next Steps

1. Finish the ML Model for MQ3's AR Application (100+ scans)
2. Create MR Application for MQ3
3. Unity Profiler Analyzer trials for both AR + MR applications
 - a. We hope to get around ~20 trials for each of these
4. VTURCS
5. Workshop Paper

Questions

1.