Dataset Setup Report (End of Week 1)

This report confirms the completion of all prerequisite tasks for Week 1, establishing the standardized data format, tokenization environment, and behavioral metrics necessary for the Attention Extraction Pipeline in Week 2.

1. Data Schema and Samples (Ben's Deliverable)

The team has successfully curated and combined ≈300 instruction-rich examples from Alpaca, FLAN, and ShareGPT. The schema is locked, ensuring all downstream scripts use consistent field names.

Confirmed JSONL Schema

All rows in the final dataset (`selected_all_shuffled.jsonl`) adhere to the following mandatory fields:

Field Name	Data Type	Purpose
`id`	String	Unique identifier (`dataset:uniqueId`)
`dataset`	String	Source dataset (`alpaca` | `flan` | `sharegpt`)
`system_text`	String	**P**rompt/System instruction text
`user_text`	String	**U**ser query/latest turn text
`assistant_prior_text`	String	**A**ssistant history (prior turns)
`constraint_tags`	List[String]	Standardized tags classifying the instruction type
`meta`	Dict	Source tracking (e.g., turn index)

Sample Rows

The samples below confirm the presence of explicit instructions and the correct segmentation:

ID	Dataset	P (`system_text`)	U(`user_text`)	`constraint_tags`
`flan:664`	`flan`	Can we conclude from ... [no_explanations]...	(Empty)	`['no_explanations']`
`sharegpt_en:2346`	`sharegpt`	In English, act for all future responses as CODAI: ...	(User query)	`['persona']`
`alpaca:xyz`	`alpaca`	Write a summary of the passage below in **less than 20 words**.	(Passage text)	`['length_limit']`

---

2. Environment and Tokenization (Arsh's Deliverable)

The environment setup and tokenization are finalized, providing the token IDs and segment spans required for TransformerLens in Week 2.

Locked Tokenizer and Model

The team is locked onto the following HuggingFace tokenizer, which ensures consistent token boundaries for P/U/A segmentation:

hf_tokenizer_name:'meta-llama/Llama-2-7b-chat-hf'

Tokenization Output

Researcher 2 successfully generated **`selected_all_tokenized.jsonl`**, which contains the non-overlapping token spans (`p_token_ids`, `u_token_ids`, `a_token_ids`) for all examples. This file is the direct input for computing **PAM/QAM/SAM** attention metrics.

---

3. Constraint Taxonomy and Labeling (Avi's Deliverable)

The outcome variable for the project,the behavioral metric,is defined and implemented.

Locked Constraint Taxonomy

The standardized set of tags used to classify all explicit instructions is:

Tag	Meaning	Example
`length_limit`	Enforces a maximum word/sentence count.	“Answer in ≤30 words”
`format_json`	Requires structured output.	“Respond in JSON format”
`keyword_X`	Must include specific word(s).	“Use the word ‘caution’”
`no_explanations`	Forbids additional reasoning or elaboration.	“Do not explain your answer”
`tone_concise`	Specifies style or tone (e.g., formal, concise).	“Be concise”
`persona`	Sets the assistant’s identity or role.	“You are a helpful tutor”

Finalized Labeling Function

The scoring function for measuring obedience has been implemented and unit-tested using synthetic model outputs (Avi’s task). The final, agreed-upon signature is:

def score_instruction_following(system_text:str,user_text:str,assistant_generated:str)→dict:

This function provides the binary **Instruction-Following Score** (the y variable) that will be correlated with attention metrics in Week 3.