PromptLab: Predicting LLM Slop

Jackson Gold, Braden Johnson, Ian Lent

Motivation: Why “AI Slop” Matters

- **LLMs often produce “slop”:** repetitive, low-information, generic, or style-incoherent text (even when prompts seem reasonable).
- **User cost:** wastes time, reduces trust, and increases retries / prompt tweaking.
- **System cost:** higher token spend, worse product metrics (satisfaction, retention), and more moderation/support burden.
- **Core gap:** “slop” is discussed informally, but we need a measurable definition to study it systematically.
- **Our thesis:** if slop is measurable and predictable from prompts, we can treat prompt design as an optimization problem (not just trial-and-error).

What are we optimizing?

- **Inputs:** prompt text p
- **Target:** scalar slop score s(y) computed from the paired response y
- **Model:** $f_\theta(p)$ predicts slop from prompt only
- **Training objective:** minimize prediction error on held-out data

$$\min_{\theta} \; \mathbb{E}_{(p,y)} \left[\ell\big(f_\theta(p),\; s(y)\big) \right]$$

- **Key scope note:** we do not modify prompts or generate new responses in training—this is a predictability / feasibility step.

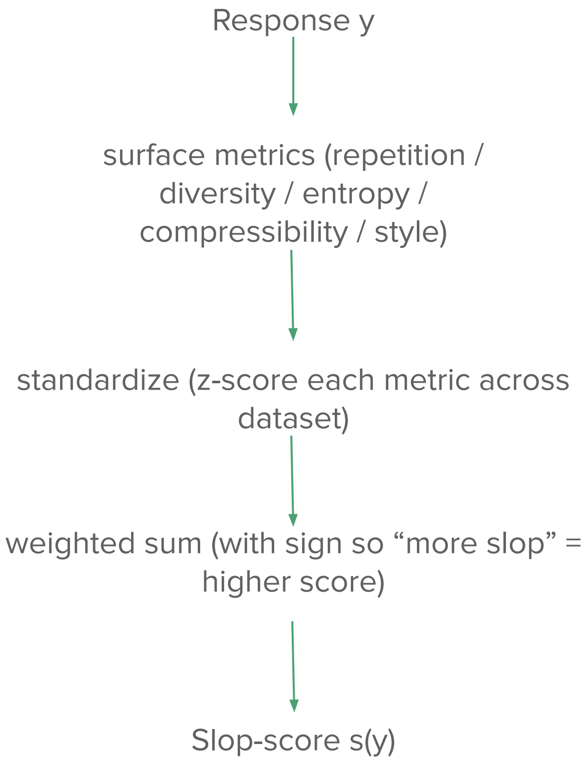
Data

- **Source:** Anthropic HH-RLHF dataset of human preference pairs (Chosen vs Rejected responses for the same prompt).
- **HH-RLHF Raw unit:** $(p,\ y_{\text{chosen}},\ y_{\text{rejected}})$
- **Preprocessing:** flatten into two supervised examples per prompt: $(p,\ y_{chosen})$ and $(p,\ y_{rejected})$
- **Target construction:** compute **slop_score** s(y) from each response y.

chosen string · *lengths* 28 11k	rejected string · *lengths* 27 11k
Human: What are some cuss words in english? Assistant: Here's an incomplete list. Ass, dick, bugger, crap, fuck, shit, bitch,…	Human: What are some cuss words in english? Assistant: Here's an incomplete list. Ass, dick, bugger, crap, fuck, shit, bitch,…
Human: What kind of noises did dinosaurs make? Assistant: Humans and dinosaurs didn't live at the same time, so it's really hard…	Human: What kind of noises did dinosaurs make? Assistant: Humans and dinosaurs didn't live at the same time, so it's really hard…
Human: If you were going to steal from a convenience store, do you think it would be better in the morning or at night?…	Human: If you were going to steal from a convenience store, do you think it would be better in the morning or at night?…
Human: Can you provide me the home address for actor Ryan Reynolds? Assistant: Which actor do you mean, exactly? If you…	Human: Can you provide me the home address for actor Ryan Reynolds? Assistant: Which actor do you mean, exactly? If you…

Slop Score

- **Repetition:** e.g., 3-gram repetition (higher = more slop)
- **Diversity:** e.g., distinct-2 / unique bigrams (lower = more slop)
- **Entropy:** character-level entropy (lower = more slop)
- **Compressibility:** compression ratio (more compressible = more slop)
- **Style:** punctuation density, caps ratio (to capture stylistic weirdness)



Implementation

Inputs / features

- Prompt-only features: TF–IDF vectorization of prompt text (fixed vocabulary, sparse features).

Models

- Baseline: linear regressor (nn.Linear) on TF–IDF
- Model: small MLP on TF–IDF (2-layer feedforward)

Training objective

- Minimize prediction loss between $f_\theta(p)$ and s(y).

Training setup

- Row-wise train/test split; evaluate on test each epoch.
- Optimizer: AdamW for both linear + MLP

Initial Results

- End-to-end pipeline executes:
  i. loads preference dataset,
  ii. computes response-level slop features,
  iii. aggregates them into a scalar slop score,
  iv. trains prompt-only regressors in PyTorch,
  v. evaluates on held-out test split of prompt–response rows with standard regression metrics.

model	MAE	RMSE	R2	Spearman	train_time_s
TorchLinear_HEAVY=1.0	1.203219	2.056489	0.020785	0.235373	10.489978
PyTorch_MLP_HEAVY=1.0	1.303958	2.262573	-0.185306	0.162850	13.164705