CIS 5300 Milestone 2

Zhirui Bian, Qianyue Ding, Maya Gambhir, Andy Li

November 19, 2025

1 Evaluation Measure

We evaluate narrative story continuation using a rubric-based LLM-as-a-judge approach on the ROCStories dataset. Our evaluation pipeline employs GPT-4.1-mini to assess model-generated continuations across six dimensions on a 0-5 scale: information completeness, factual accuracy, relevance, logical coherence, creativity expression, and overall quality.

The primary metric is the **OverallScore**, formally defined as:

$$\text{OverallScore}(S) = \frac{1}{N}\sum_{i=1}^{N}[\text{overall\_raw}(x_i, y_i) + \text{length\_bonus}(y_i)]$$

where N is the number of examples (500 in our test set), overall_raw is the judge's direct assessment, x_i is input prompt (including beginning + prompt_type instructions), y_i is model-generated continuation (model answer), and length_bonus is the length-based adjustment. For training reward models with 0-1 scale, this score undergoes normalization by dividing by 5.

The evaluation script (`judge_llm.py`) provides the judge with the story beginning, reference ending (as guidance only, not ground truth), and model-generated continuation. For structured prompts (moderate/strict), a small controlled smoothing factor based on output length is applied to prevent penalization of well-formed template responses, capped to avoid dominating the evaluation.

1.1 Length Bonus Rationale

Producing longer outputs with more reasoning steps while maintaining high accuracy is inherently more difficult. Shorter outputs have lower error probability because they expose fewer details that could be wrong. This fundamental asymmetry necessitates a length bonus/penalty system.

For moderate and strict prompts, we care about both correctness and explicit reasoning. Extremely short answers tend to expose fewer intermediate steps and therefore have a lower chance of making visible mistakes, but they also provide less evidence of genuine reasoning. To mildly encourage well-developed, step-by-step explanations that remain accurate, we add a small length-based bonus to the judge's overall score for answers longer than 80 words (capped at +1.0 and clipped at a maximum score of 5).

The bonus structure is:

- Under 80 words: no bonus
- 80-160 words: linear bonus scaling
- Over 160 words: capped at +1.0

This does not reward verbosity alone: short but correct answers keep their original score, and overly long answers can only gain at most +1.0 point on top of the judge's raw overall quality score.

2 Baselines and Performance

2.1 Simple Baseline: Direct Generation

Our simple baseline uses DeepSeek-R1-Distill-Llama-8B (4-bit quantized) with loose prompting that requests only a story continuation without reasoning steps. This represents a "non-rationalized direct decode baseline" where the model generates continuations without explicit chain-of-thought reasoning. The prompt simply provides the story beginning and asks for a plausible ending.

2.2 Strong Baseline: Structured Chain-of-Thought

The strong baseline employs the same model but with structured chain-of-thought (CoT) prompting. We evaluate two prompt variants:

- **Moderate CoT**: Includes reasoning guidance with flexible structure
- **Strict CoT**: Enforces a four-step template (Setup/Reasoning/Check/Answer)

These prompts explicitly instruct the model to reason through narrative coherence before generating the continuation, with the hypothesis that structured reasoning will improve logical coherence and reduce hallucination compared to direct generation.

2.3 Results

Table 1 and Figure 1 present the evaluation results on our 500-example test set across all rubric dimensions. The strong baseline with structured CoT prompting shows substantial improvements across all metrics.

2.4 Analysis

The structured CoT approach yields a +0.45 improvement in overall quality compared to the simple baseline, representing a 13.8% relative gain. Notable improvements in factual accuracy (+0.45) and relevance (+0.40) suggest that explicit reasoning steps help the model maintain narrative consistency and avoid off-topic content.

The strict CoT template slightly outperforms the moderate version, indicating that more structured reasoning constraints can benefit coherent story generation. However, the improvement is marginal (3.70 vs 3.65), suggesting diminishing returns from overly rigid templates.

These baselines establish a strong foundation for future work. The consistent improvements from CoT prompting validate our hypothesis that structured reasoning reduces hallucination in narrative generation. Future milestones will explore reward modeling (ORM/PRM) and reinforcement learning approaches to further improve performance beyond these supervised baselines.

3 Implementation Details

All experiments use DeepSeek-R1-Distill-Llama-8B with 4-bit quantization for local inference. Evaluation employs GPT-4.1-mini as the judge model. The complete pipeline includes:

- `build_prompts.py`: Generates three prompt variants per story
- `judge_llm.py`: Implements rubric-based evaluation
- `simple-baseline.py`: Direct generation without CoT
- `strong-baseline.py`: Structured CoT generation

Full scoring details and rubric definitions are provided in `scoring.md`. The evaluation script can be executed as:

```
python -m app.roc_eval.judge_llm --input narrative_with_prompts.csv \
    --output narrative_scores.csv --openai_model gpt-4.1-mini
```

Table 1: Baseline Performance on ROCStories Test Set (N=500)

Model	Info. Comp.	**Factual Acc.**	**Relevance**	Logical Coh.	Creativity	Overall Raw	Overall Score
Simple (loose)	1.734	**4.988**	**3.610**	2.852	2.764	2.556	2.556
Strong (moderate)	1.730	**4.996**	**3.604**	2.754	2.086	2.314	3.650
Strong (strict)	2.532	**4.962**	**4.354**	3.506	2.480	2.956	**3.700**

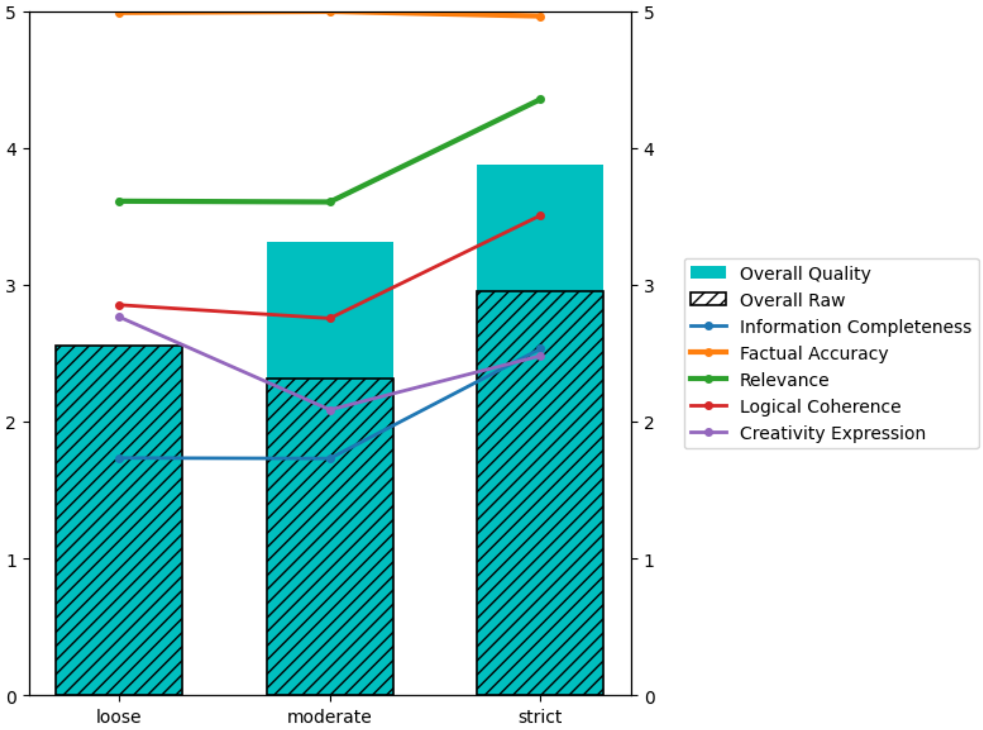

Figure 1: Performance metrics across prompt types. Bar chart shows Overall Quality (solid) and Overall Raw (hatched). Lines indicate individual rubric dimensions.