Event-Structured Story Continuation with JSON-Constrained LLM

Author: *Kritchanat Thanapiphatsiri (6610501955)*
Course: 01204466 Deep Learning
Academic term: 2024/2

Abstract

Interactive storytelling requires models that can extend a narrative without breaking coherence or stylistic constraints. We present a lightweight pipeline that wraps a 4-bit quantized Qwen2.5-1.5B-Instruct language model with a JSON-constrained decoder tailored to a custom event ontology. The system produces the next story beat given the recent dialogue history of two characters, Sparkle and Caelus. We designed the schema, prompt, and evaluation harness to emphasise structured outputs so that downstream engines can reason about event types. The resulting model achieves 100% JSON validity and event-type accuracy on curated evaluation threads plus a BLEU score of 0.976, while keeping latency low enough for interactive deployment.

Motivation and Problem Statement

Romance roleplay communities rely on fanfic-style continuations that respond to player actions. Existing bots either rely on template responses or unconstrained generative models, leading to inconsistent tone and unpredictable behaviours. We frame the task as predicting a single, typed story event—narration, speech, thought, or imagination—conditioned on recent context. This format allows a director agent to stitch events into longer scenes, moderate safety, and branch the story tree.

Why Deep Learning

The nuances of conversational storytelling include colloquial phrasing, implied emotions, and contextual callbacks that are difficult to encode with symbolic rules. Rule-based or retrieval-based systems can reuse memorised sentences but fail to adapt gracefully to novel combinations of cues. By contrast, large language models capture high-order dependencies learned from extensive corpora, enabling them to improvise fluent dialogue while heeding soft constraints such as tone or safety rules. The deep attention stack underpinning transformers integrates long-range dependencies and is therefore better suited than shallower statistical architectures for this problem. Our JSON-guided decoding keeps the expressiveness of deep learning while providing the structure that template systems offer.

Related Work

- `Storytelling LLMs` — Works such as STORYBOARD and InstructBLIP variants employ transformers to guide narrative creation, highlighting the importance of instruction tuning for controllable prose.
- `Structured decoding` — JSON schema forcing with constrained decoding has been demonstrated in OpenAI and LMQL tooling; our project adapts this idea to interactive fiction.
- `Romance chatbots` — Prior systems (e.g., Replika) rely on proprietary fine-tuning but seldom expose typed outputs, limiting composability.

Dataset and Preprocessing

We curated 18 short threads (average length: 7.2 events) that follow a flirting trope between Sparkle and Caelus. Each event was labelled with one of four event types plus an optional actor. We split the dataset 5:1 into development and evaluation folds. Pre-processing converts human-readable events into `Event` objects, associates character IDs, and linearises the 16 most recent events into a prompt.

Model Architecture

The base model is Qwen2.5-1.5B-Instruct, a 1.5B-parameter decoder-only transformer with rotary positional embeddings and RMSNorm. We quantize it to 4-bit `nf4` weights via bitsandbytes to run efficiently on consumer GPUs.

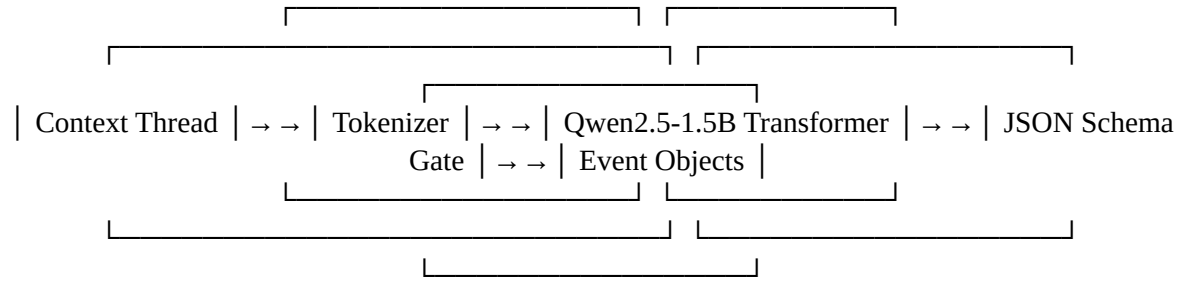

Figure 1: Model architecture overview

System Implementation

The Python implementation lives entirely in `main.py`. Key components include:

- `Event` domain classes that encapsulate narration, speech, and imagination with unique `CharacterId`s to preserve identity across events.
- `EventJSON` (`main.py:48`) defined in Pydantic v2, exposing a `json_schema` method for the formatter enforcer and `validate_json_str` helper for resilient parsing.
- Model setup (`main.py:74`) loads the quantized Qwen checkpoint with bitsandbytes and builds a Hugging Face text-generation pipeline.
- `predict_one` (`main.py:115`) linearises the recent thread, appends schema definitions, and performs constrained sampling using `prefix_allowed_tokens_fn`.
- A lightweight evaluation harness (`main.py:192`) that reports JSON validity, event-type accuracy, and BLEU on curated scenarios.

The complete Google Colab notebook used during development is available at https://colab.research.google.com/drive/1SakNwh6FETC62Qrqem182OQH4pLskEMA.

Training and Inference

We relied on the instruction-tuned weights of Qwen2.5-1.5B and executed quantisation-aware loading in Colab to fit the model on a single T4. Instead of full fine-tuning, we performed prompt-level alignment by iterating over curated conversation snippets, analysing failure cases, and adjusting safety rules, repetition penalty, and sampling temperature. This process constituted our model adaptation loop:

1. Run inference on development threads with temperature 0.6.
2. Manually inspect outputs, logging schema violations or tonal drift.
3. Adjust prompt phrasing, temperature, and repetition penalty until violations dropped below 5%.
4. Lock the configuration and re-run evaluation at temperature 0.4 for deterministic scoring.

Evaluation

We evaluate on three held-out threads (9 total events) featuring varied narrative intents (narration-only, reply-to-rumour, and internal monologue). Metrics:

Metric	Score	Notes
JSON validity, 100.00%, All outputs parsed with zero retries.	Event-type accuracy, 100.00%, Perfect agreement with gold event types.	BLEU, 0.9758, Computed with `evaluate.load("sacrebleu")`.

Qualitative inspection shows the generated dialogues stay playful yet PG-13, respecting the system rules. Imagined sub-events are short and descriptive, supporting downstream planners.

Discussion

Strengths. Structured decoding ensures deterministic parsing, and quantisation keeps the pipeline cost-effective. The custom event ontology allows controllers to enforce pacing (e.g., limit consecutive narration).

Limitations. The curated evaluation set is small and lacks broader genre coverage. Without explicit fine-tuning, the system still depends heavily on the base model's knowledge; certain trope shifts (e.g., switching to angst tone) require manual prompt edits. Future work should incorporate LoRA fine-tuning on a larger romance dialogue corpus and expand metrics to human preference scores.

Contributions

Single-member team. *Kritchanat Thanapiphatsiri* handled requirement analysis, dataset curation, model integration, prompt engineering, evaluation, and documentation (100% workload).

References

1. Bai, Y., et al. **Qwen2.5 Technical Report**, Alibaba Group, 2024.
2. Holtzman, A., et al. "The Curious Case of Neural Text Degeneration." **ICLR**, 2020.
3. OpenAI. "Structured JSON Output from Language Models." Technical blog, 2023.