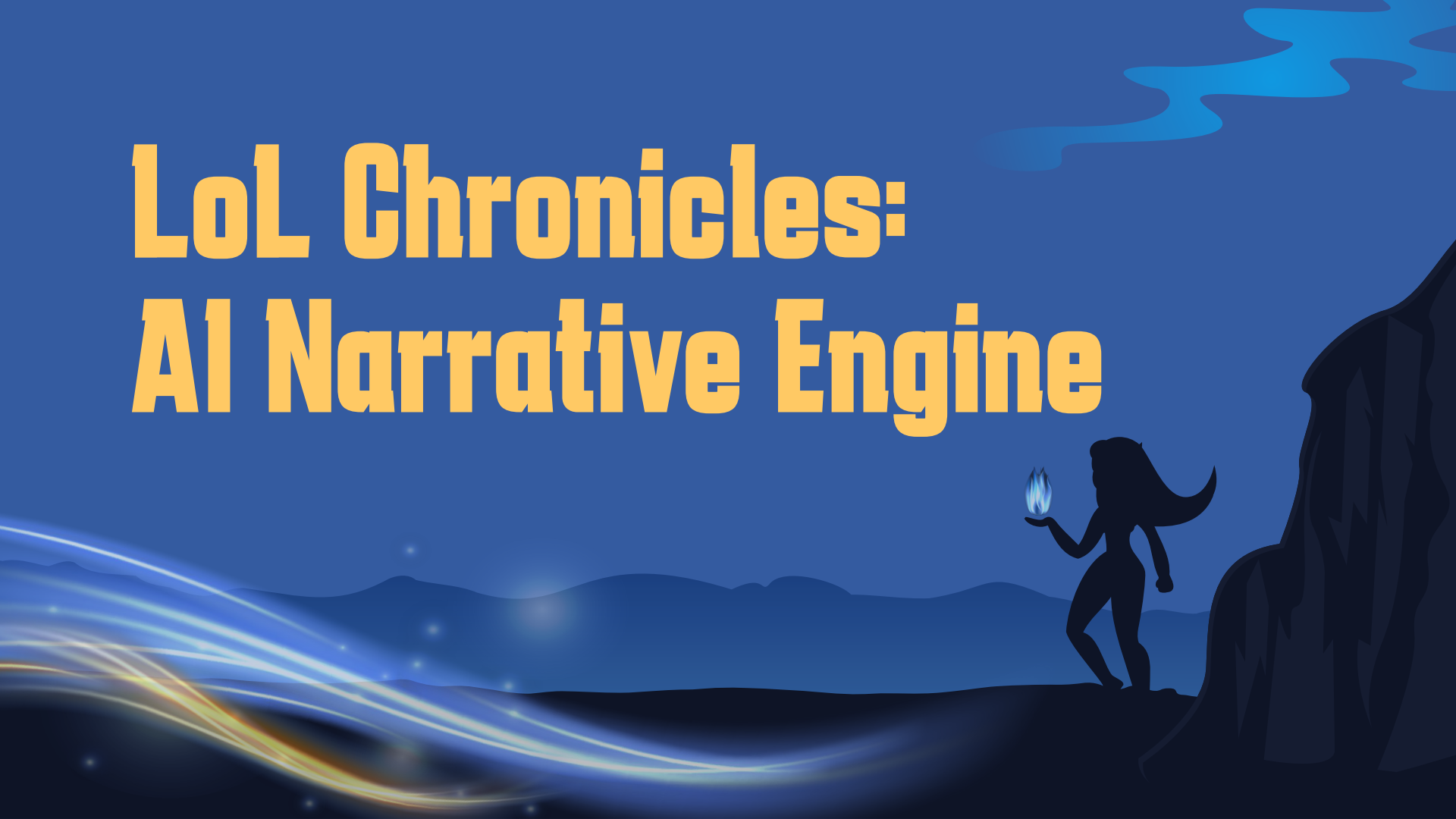


LoL Chronicles: AI Narrative Engine



Our Objective

Problem Statement: Creating consistent, engaging content for 170+ champions is time-consuming and requires deep lore knowledge

Core Objectives:

1. Develop a specialized language model for League of Legends narrative generation
2. Create contextually appropriate content consistent with character personalities
3. Build a tool for efficient lore-consistent content creation



The background is a deep blue gradient. On the left, a black silhouette of a person in a crouched position holds a glowing, futuristic device. Bright, curved light trails emanate from the device, extending towards the upper left. The bottom of the image features a dark, jagged silhouette of a horizon or mountain range. In the upper right, there are lighter blue, cloud-like shapes.

01

DATA COLLECTION

Using Web Scraping for LoL Champion Data

Approach



ESSENTIAL DATA

Name, region, role,
race, quote, short bio,
related champions,
biography, story



WHY SCRAPING?

Structured data not readily
available, wanted more
accurate information directly
from the official lore



TOOLS

Python and Selenium

Challenges and Solutions

Challenge A

Dynamic Content Loading

Solution: Selenium for browser automation that is JavaScript-based

Challenge B

Inconsistent HTML

Multiple CSS selector strategies with graceful fallbacks



170 Champions Successfully Scraped!

```
[
  {
    "name": "AATROX",
    "url": "https://universe.leagueoflegends.com/en_US/champion/aatrox/",
    "region": "RUNETERRA",
    "role": "FIGHTER",
    "race": "DARKIN",
    "quote": "\"I must destroy even hope...\"",
    "short_bio": "Once honored defenders of Shurima against the Void, Aatrox and his brethren would eventual",
    "related_champions": [
      "Kayn",
      "Varus",
      "Tryndamere"
    ],
    "bio_url": "https://universe.leagueoflegends.com/en_US/story/champion/aatrox/",
    "story_url": "https://universe.leagueoflegends.com/en_US/story/aatrox-color-story/",
    "full_biography": "Whether mistaken for a demon or god, many tales have been told of the Darkin Blade...",
    "full_story": "Darkness.\n\nThe breath I cannot take plagues me.\n\nIt is an emptiness in my lungs and t
  },
  {
    "name": "AHRI",
    "url": "https://universe.leagueoflegends.com/en_US/champion/ahri/",
    "region": "IONIA",
    "role": "MAGE",
    "race": "VASTAYA",
    "quote": "\"Human emotions can be more volatile than even the deepest magic.\"",
    "short_bio": "Innately connected to the magic of the spirit realm, Ahri is a fox-like vastaya who can ma",
    "related_champions": [
      "Yasuo",
      "Lillia",
      "Wukong"
    ],
    "bio_url": "https://universe.leagueoflegends.com/en_US/story/champion/ahri/",
    "story_url": "https://universe.leagueoflegends.com/en_US/story/ahri-color/",
    "full_biography": "For most of her life, Ahri's origins were a mystery to her, the history of her vastay",
    "full_story": "The market smelled of burning incense and rotting cabbage.\n\nAhri wrapped her cloak arou
```

02

Fine-tuning

Training a Specialized Model for LoL Narratives



Fine-tuning Steps

01

Model Selection

Select the Base
Model

02

Data Tokenization

Tokenize the
Scraped Data

03

Model Fine-tuning

Select Hyperparameters and
Fine-tune in the Cloud

Model Selection: GPT2-XL

GP2-XL is our base model for fine-tuning:

- **Parameters:** 1.5B
- **Source:** Hugging Face

Rational for model selection:

1. Open-weight & permissive MIT license
2. Decoder-only architecture for story generation
3. Trainable locally



Tokenization

Goal:

- Transform rows of fine-tuning text data (1 story per row) into rows of **Byte-Pair Encoding** and **Attention Mask**

```
{  
  'text': 'Ahri whispered an ancient spell...'  
}
```



```
{  
  'input_ids': tensor([502, 345, ..., 50256, 50256]),  
  'attention_mask': tensor([ 1, 1, ..., 0, 0])  
}
```

Tokenization

- Loads the original tokenizer
- Use end-of-sentence token (ID 50256) for padding
- Tokenization consideration
 - Truncate at 1024 tokens
 - Pad shorter inputs to 1024 tokens
- Tokenize

```
tokenizer = AutoTokenizer.from_pretrained(
    "models/gpt2-xl"
)
tokenizer.pad_token = tokenizer.eos_token

ds = Dataset.from_dict({"text": texts})

# GPT-2's max context length is 1024
context_length = 1024

def tokenize(batch):
    return tokenizer(
        batch["text"],
        truncation=True,
        max_length=context_length,
        padding = "max_length"
    )

ds_tok = ds.map(
    tokenize, batched=True, remove_columns=["text"]
)
ds_tok.set_format(
    type="torch", columns=["input_ids", "attention_mask"]
)
ds_tok
```

Fine-tuning

Goal:

- Change model weights to generate better stories for League of Legends

Hyperparameters for Best Performance

- `per_device_train_batch_size` = 1
- `gradient_accumulation_steps` = 8
- `Learning_rate` = $5e-5$
- `Epochs` = 3

Fine-tuning

1. Forward Pass

- Embedding (B, T, D)
- 25 attention heads/block
 - Causal masking
- 48 blocks
- Logits layer

2. Loss Calculation

$$\mathcal{L}_{\text{batch}} = \frac{1}{B} \sum_{i=1}^B \mathcal{L}_i$$
$$= -\frac{1}{B} \sum_{i=1}^B \log \left(\frac{\exp(z_{i,c_i})}{\sum_{k=1}^{|V|} \exp(z_{i,k})} \right)$$

3. Backward Pass

- Loss gradient
- Chain rule all the way back
- Store gradients



Every 8 Cycles

Optimizer Step: weights updated

Final Loss: 2.45

Fine-tuning - Cloud Setup



Modal

- Create an Modal app
- Set up an image for the container
- Specify GPU type,
- Set up volumes (mount data & save model artifacts)
- **Deploy - up to 10 concurrent H100s**

```
fine_tune_app = modal.App("gpt2-xl-ft")
ft_image = modal.Image.debian_slim().pip_install(
    "transformers", "datasets", "accelerate",
    "bitsandbytes", "peft", "torch"
)
GPU = "H100"
vol = modal.Volume.from_name("data")
model_vol = modal.Volume.from_name(
    "gpt2_ft", create_if_missing=True
)

@fine_tune_app.function(
    image=ft_image,
    gpu=GPU,
    timeout = 3600,
    volumes={"/data": vol, "/checkpoints": model_vol}
)
def train_model():
```

The background is a dark blue gradient with stylized, lighter blue cloud-like shapes at the top. A bright sun or moon with a lens flare is positioned behind the number '03'. On the right side, there is a silhouette of a person in a running or jumping pose, holding a torch that is lit with a bright yellow and orange flame.

03

Evaluation

Using LLM-as-a-Judge for Evaluation

Model Evaluation: LLM as a Judge

Gemini 2.0
Flash

01

Coherence

Logical Flow

02

Relevance

Stay on Topic

03

Narrative

Engaging & Well
Written

04

Originality

Novelty &
Creativity

Story Generation Prompt

```
# 5 prompts
prompts = [
    "Ashe, the Frost Archer, stalks her prey in the icy forests of Freljord. Continue the story:",
    "Zed, the Master of Shadows, prepares his ultimate strike at dusk. Continue the tale:",
    "Soraka, the Starchild, descends to heal the wounded on the Fields of Justice. Continue:",
    "Dr. Mundo, the Madman of Zaun, rampages through the chem[ ]labs. Continue:",
    "Jinx, the Loose Cannon, plots her next explosive prank in Piltover. Continue:"
]
```

Base Story

How it Began: The Night of Nightmares is a three-part short-story adventure for 5e D&D, written by Jonathan Tweet.

There are no monsters or PCs in this episode—the plot kicks off in a tavern, where the party has been hired by a wealthy merchant to retrieve a shipment of goods.

As they explore the dark warehouse, the ship's cargo proves far more sinister than anyone expected.

Result:

The baseline model drifts into tabletop-RPG promo text instead of League lore.

Fine-Tuned Story

Zed the Illuminator

Trundle's apprentice, Zed the Illuminator, was destined to master the dark arts—yet in a desperate quest to save the dying world of Shurima, he unleashed an arcane plague that turned its denizens into bloodthirsty monsters.

For centuries, Zed studied and honed the ancient magic of shadow.

His master's instructions were clear: draw no attention, invite no gaze of the living.

Such was the decree of the elders' great teacher, Master Trundle.

Result:

Emphasize lore consistency: Shurima setting, Trundle reference, shadow-magic tone – clear improvement over baseline.

Evaluation Generation Prompt

```
instruction = (  
    "You are a story evaluator. For the given input prompt and its generated "  
    "story, rate each of the following on a scale of 1 (worst) to 5 (best) "  
    "and return ONLY a JSON object with keys:\n"  
    '  "coherence": ...,      # logical flow\n'  
    '  "relevance": ...,      # stays on topic\n'  
    '  "narrative_quality": ..., # engaging & well-written\n'  
    '  "originality": ...      # novelty / creativity\n\n'  
    f"Input Prompt:\n{prompt}\n\nGenerated Story:\n{story}"  
)
```

Metric	Base GPT-2	Fine-tuned	Δ
Coherence	2.4	3.7	+1.3
Relevance	2.0	4.1	+2.1
Narrative Quality	2.6	3.8	+1.2
Originality	2.1	3.5	+1.4

Limitations and Next Steps

Limitations:

- Context window constraints
- Evolving lore
- LLM as a Judge evaluation blind spots

Next Steps

- Fine-tune larger models
- Real-time lore data pipeline
- Creator-friendly tools

A stylized illustration of a dragon in silhouette, breathing a stream of fire. The dragon is positioned on the left side of the frame, facing right. The fire is depicted as a bright, glowing stream of light that curves upwards and to the right. The background is a solid blue color with some lighter blue, wavy shapes in the upper left corner, suggesting a sky or water. The overall style is minimalist and modern.

THANKS!

GitHub: [Link](#)

CREDITS: This presentation template was created
by **Slidesgo**, including icons by **Flaticon**, and
infographics & images by **Freepik**