16.12.25

Wizard of POdz: An Adaptive AI Podcast Clip Generator

Group 8: Sebastian Käslin, Kaushik Raghupathruni

Modeling and Scaling of Generative AI Systems

1. **Project Idea**
2. **System Architecture**
3. **Static Analysis**
4. **Online Optimisation**
5. **Demo**
6. **Conclusions**

Project Idea

Podcast Generation Application

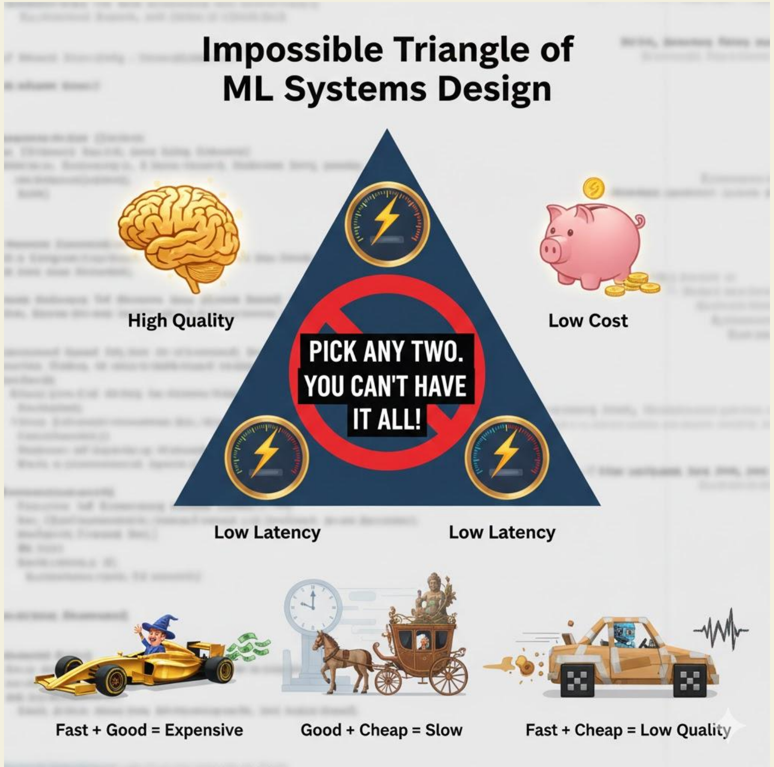
- Generate a 150-200 word (60-80 seconds) podcast introduction
- Combine LLM and two TTS (adaptive switch)
- User selects a topic
- Simulate host-guest interaction

Motivation

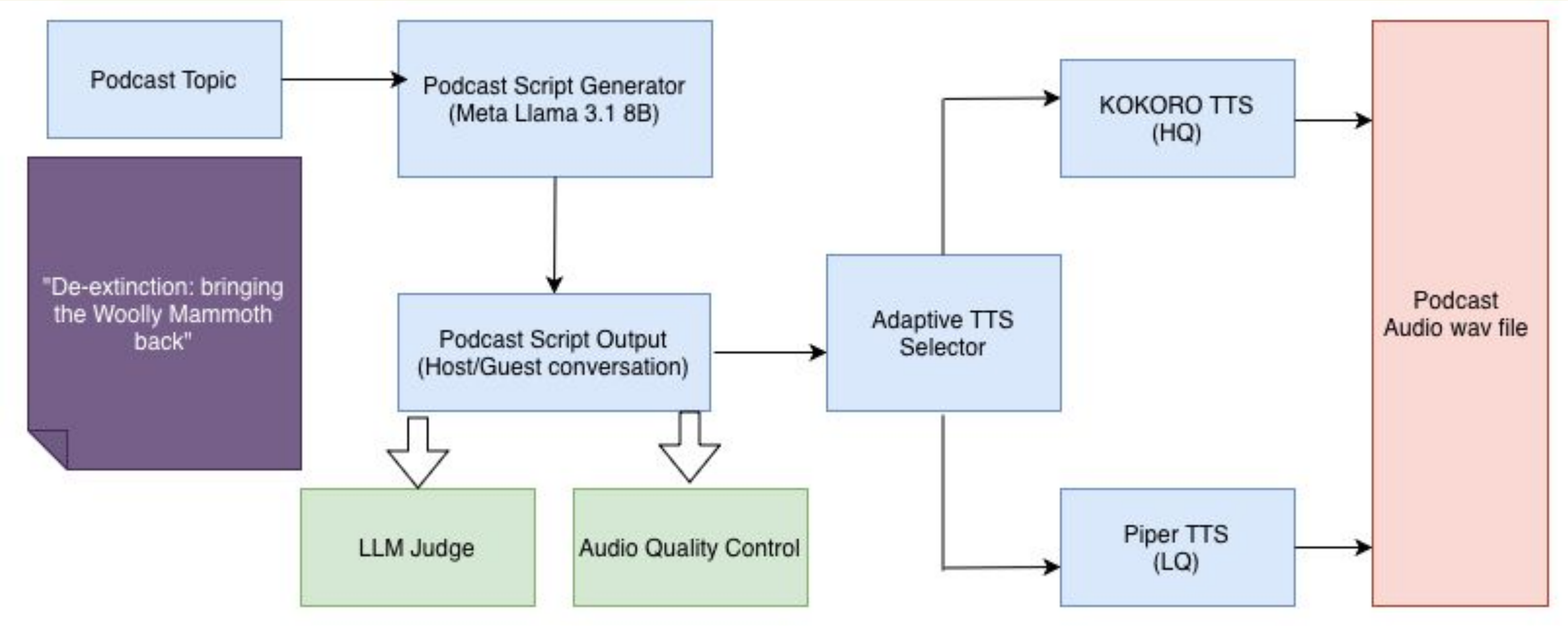
- Practical use case for orchestrating multiple AI technologies
- Two End-to-end pipelines:
  text generation -> audio synthesis

Experimentation:

- Evaluate: the quality and best model configurations (offline).
- Explore Scaling : Test pipeline behaviour under different load conditions and optimise throughput (online).


Impossible Triangle of ML Systems Design
High Quality
Low Cost
PICK ANY TWO. YOU CAN'T HAVE IT ALL!
Low Latency
Low Latency
Fast + Good = Expensive
Good + Cheap = Slow
Fast + Cheap = Low Quality

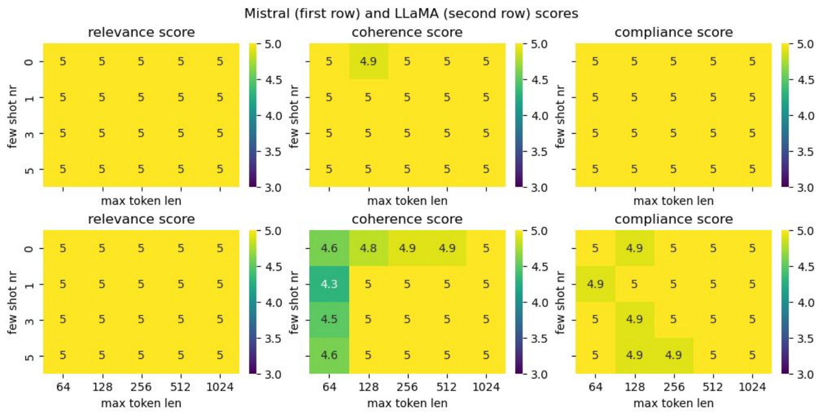
System Architecture


Podcast Topic
"De-extinction: bringing the Woolly Mammoth back"
Podcast Script Generator (Meta Llama 3.1 8B)
Podcast Script Output (Host/Guest conversation)
LLM Judge
Audio Quality Control
Adaptive TTS Selector
KOKORO TTS (HQ)
Piper TTS (LQ)
Podcast Audio wav file

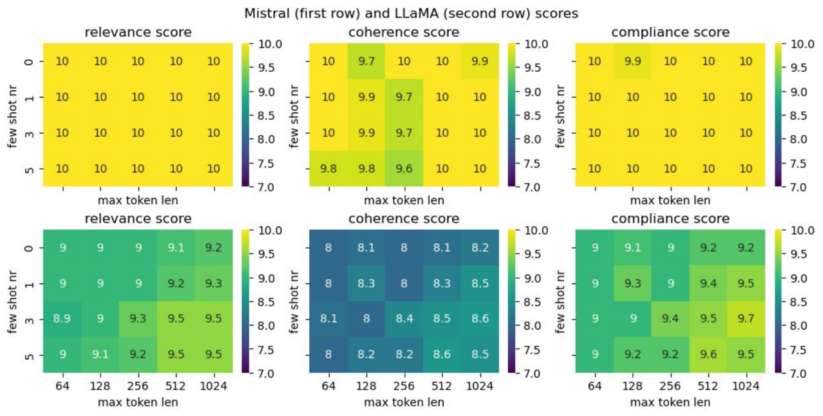
Static Analysis of LLM [Script Quality]

- Hyperparameter search:
  - 20 configs for Max Tokens len, no. of few shot examples.
- Evaluation Dataset:
  - 10 podcast topics.
  - 200 generated podcast scripts.
- Prompt Engineering
  - Produce structured podcast script (metadata + labelled dialogue)
- Quality Metric:
  - Relevance score
  - Coherence score
  - Compliance score
- Quality Evaluation method:
  - Use 2 LLMs as a Judges
    - LLaMA 3.1-8B Instruct itself
    - Mistral 7B-Instruct v0.3
  - 2 scales: [1-5] and [1-10]

Rating scale [1-5]



Rating scale [1-10]

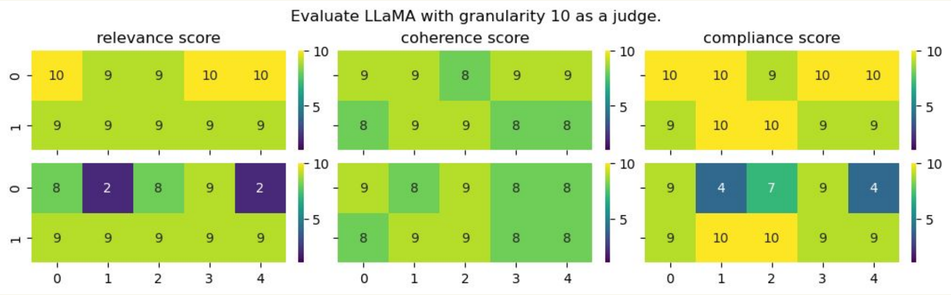


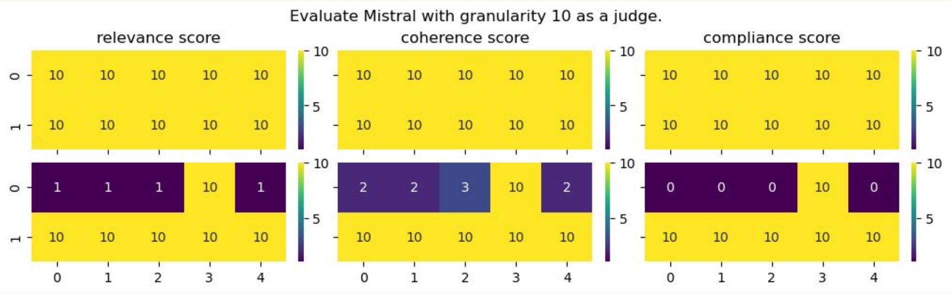
Static Analysis of LLM [Evaluate a Judge]

- Judges may be too loose or not able to produce meaningful quality metrics reliably

- Few-shot examples experiment
  - 10 few-shot examples generated by ChatGPT (quality checked)
  - Ask ChatGPT to introduce errors in the first five few shot examples
  - Passed to the judges both originals and with errors

- Conclusions
  - Our model is producing good quality podcasts (mistral high scores)
  - LLaMA is not reliable
  - Mistral is not suited to discriminate fine quality differences to choose configurations (either very bad or very good)





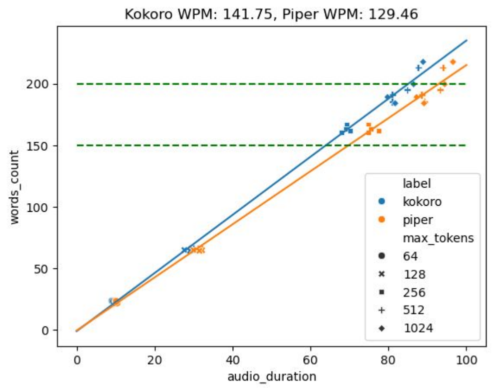
Static Analysis of the Pipeline

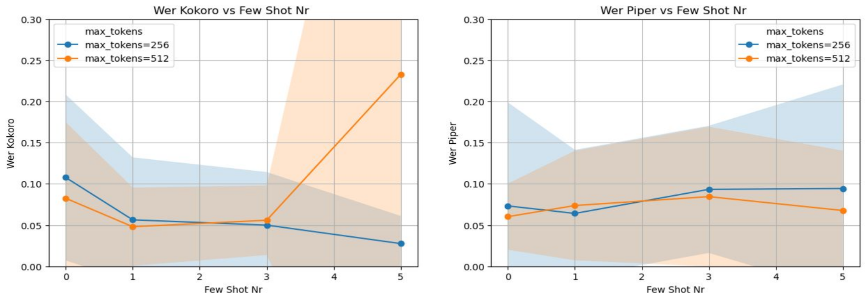
- Task Requirement
  - Produce 150-200 words podcast clip
  - 256, 512 max token lengths respect this requirement

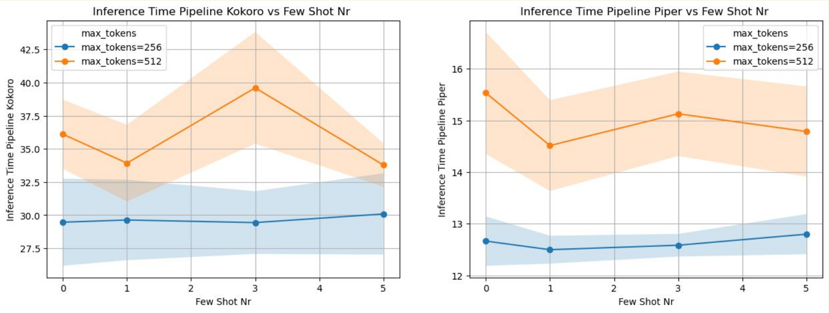
- Audio Quality Metric:
  - Word Error Rate (WER)
  - ASR Whisper-small model to create the generated podcast audio clip transcription

- Audio Quality vs Latency method:
  - Compared inference latency for LLM-Kokoro and LLM-Piper at 256 vs 512 max tokens.
  - Priority on reducing inference time (256)
  - Selected 3 few-shot examples as a trade-off between enforcing output structure and limit risk of overfitting to the few shot examples

- Chosen optimal configuration
  - (max_token_len, few_shot_nr) = (256, 3)



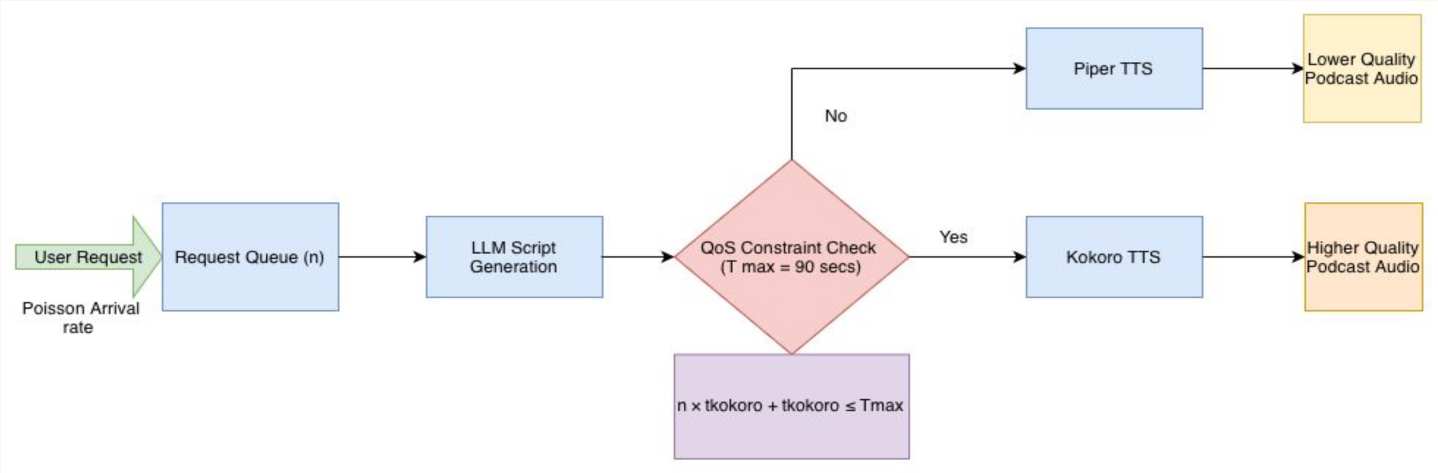




Online Analysis

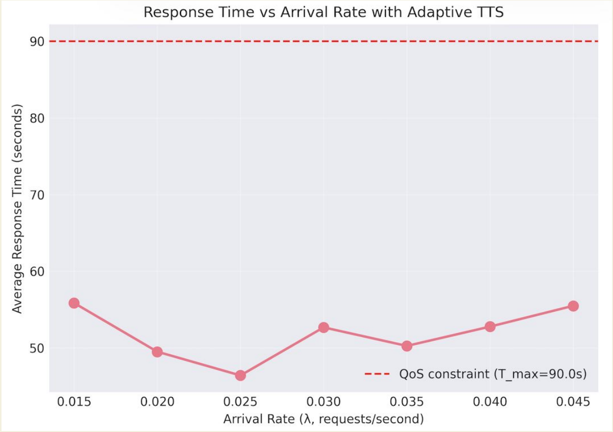
- Model:
  - We used an M/G/1 Queue with state dependent service rate.
- QoS Constraint:
  - Response time less than 90 secs.
- Adaptive Switching Policy:
  - Use kokoro only if the estimated wait time including current request is within 90 secs.
  - Otherwise switch to piper to clear queue faster.

$$\text{Backend} = \begin{cases} \text{Kokoro} & \text{if } n \times t_{\text{kokoro}} + t_{\text{kokoro}} \leq T_{\text{max}} \\ \text{Piper} & \text{otherwise} \end{cases}$$


User Request
Poisson Arrival rate
Request Queue (n)
LLM Script Generation
QoS Constraint Check (T max = 90 secs)
n × tkokoro + tkokoro ≤ Tmax
No
Yes
Piper TTS
Kokoro TTS
Lower Quality Podcast Audio
Higher Quality Podcast Audio

Online Analysis

- <u>Experimental setup</u>
  - 7 Arrival rates λ from 0.015 to 0.045 req/s (54 to 162 req/hr) each 30 mins duration.
  - Range is selected to test light to heavy load conditions without exceeding piper's max capacity.
- <u>Stability Conditions:</u>
  - All λ < μ (0.054req/sec) system is stable.
  - Handled arrival rates up to 75.8% of Piper TTS's max capacity.
- <u>Response Time Management:</u>
  - Response time remains bounded (46-55 sec) and is below 90 sec. (QoS compliant).


Response Time vs Arrival Rate with Adaptive TTS

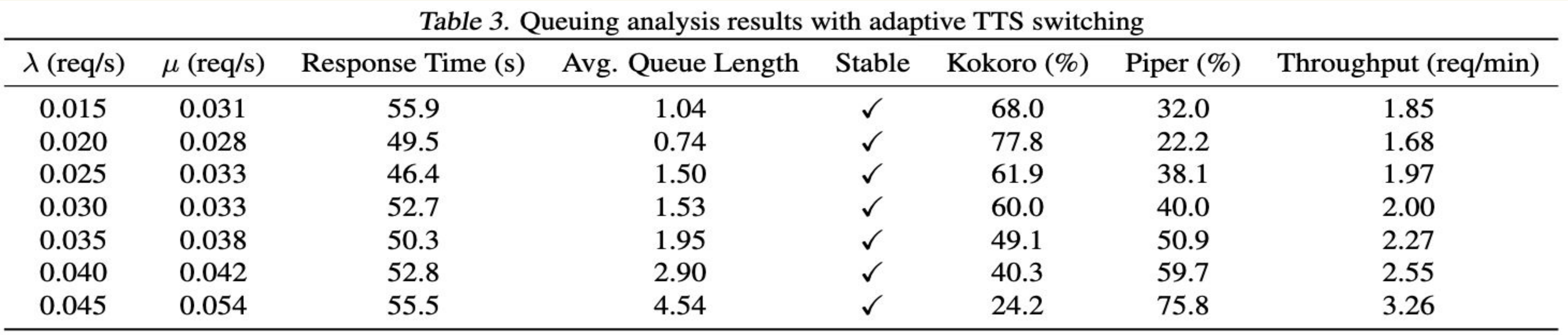


Table 3. Queuing analysis results with adaptive TTS switching

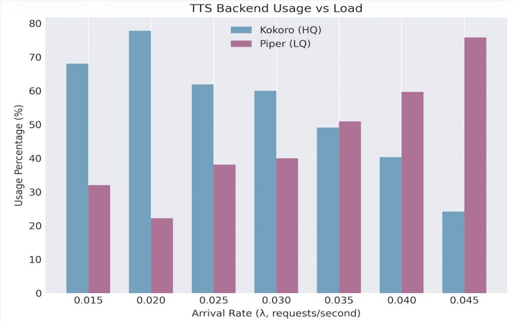
λ (req/s)	μ (req/s)	Response Time (s)	Avg. Queue Length	Stable	Kokoro (%)	Piper (%)	Throughput (req/min)
0.015	0.031	55.9	1.04	✓	68.0	32.0	1.85
0.020	0.028	49.5	0.74	✓	77.8	22.2	1.68
0.025	0.033	46.4	1.50	✓	61.9	38.1	1.97
0.030	0.033	52.7	1.53	✓	60.0	40.0	2.00
0.035	0.038	50.3	1.95	✓	49.1	50.9	2.27
0.040	0.042	52.8	2.90	✓	40.3	59.7	2.55
0.045	0.054	55.5	4.54	✓	24.2	75.8	3.26

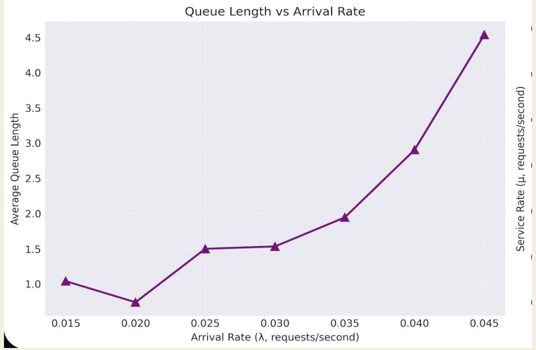
Online Analysis

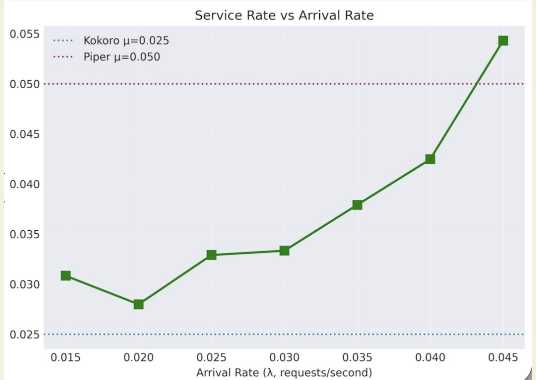
- Progressive Quality degradation:
  - As arrival rate increases, Kokoro usage decreases from 68% to 24%. While piper usage increases from 32% to 75.8%

- Queue Regulation:
  - As arrival rate increases Queue lengths grow from 1.0 to 4.5.

- Service rate adaption:
  - As arrival rate increases service rate increases. It shows how system adjusts capacity by switching to faster backend, Piper TTS.





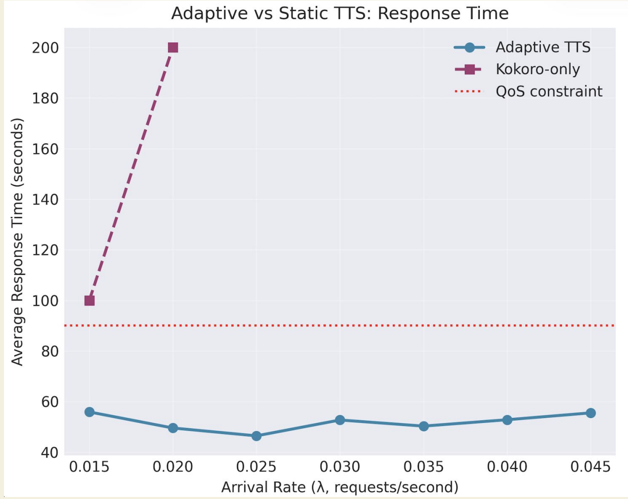


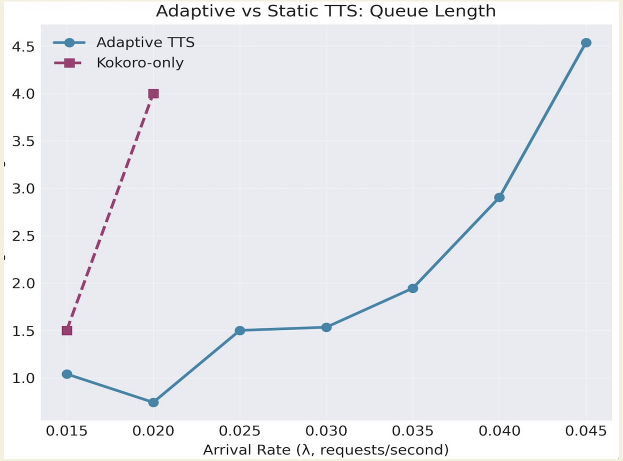
Online Analysis

- Comparison with Static Strategies:

  At arrival rate: 0.035 req/sec and simulation time duration of 30 mins:

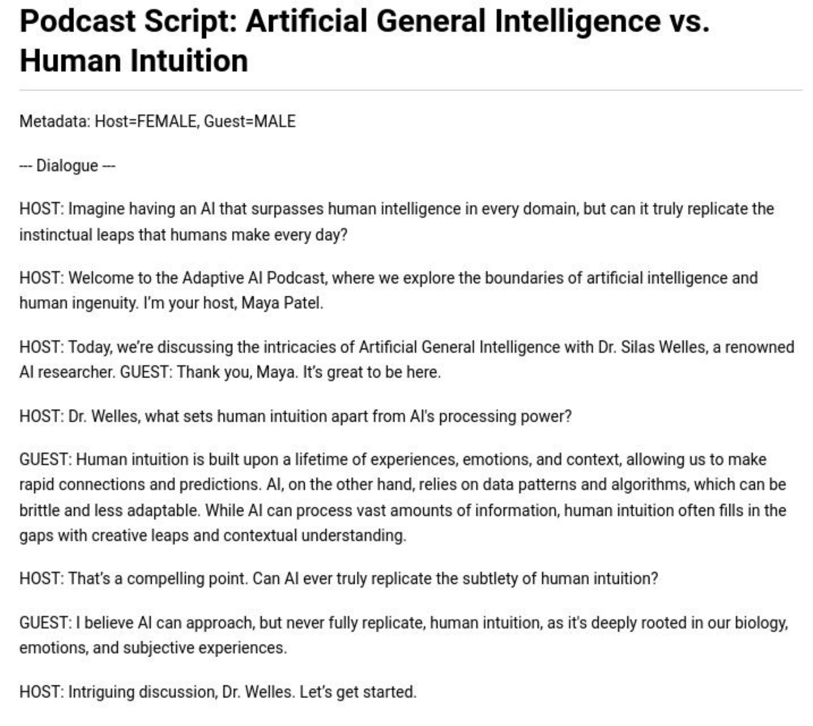
  - Adaptive Switching:
    - Response time: 50.3 secs
    - Quality: 49.1%Kokoro usage.
    - Queue length: 1.95
  - Static - Kokoro only:
    - Response time: 142.7 sec [QoS violated]
    - Quality: 100%, kokoro usage.
    - Queue length: Unbounded growth
- Limitations:
  - Poisson arrival rate assumption does not hold for all real world scenarios. [Bursty traffic patterns].
  - Switching overhead: Frequent switching between models could impact system latency.
  - Due to compute restrictions, we simulated arrival rates for only for 30 minute durations.





Demo

Podcast Script:



Podcast Script: Artificial General Intelligence vs. Human Intuition

Metadata: Host=FEMALE, Guest=MALE

--- Dialogue ---

HOST: Imagine having an AI that surpasses human intelligence in every domain, but can it truly replicate the instinctual leaps that humans make every day?

HOST: Welcome to the Adaptive AI Podcast, where we explore the boundaries of artificial intelligence and human ingenuity. I'm your host, Maya Patel.

HOST: Today, we're discussing the intricacies of Artificial General Intelligence with Dr. Silas Welles, a renowned AI researcher. GUEST: Thank you, Maya. It's great to be here.

HOST: Dr. Welles, what sets human intuition apart from AI's processing power?

GUEST: Human intuition is built upon a lifetime of experiences, emotions, and context, allowing us to make rapid connections and predictions. AI, on the other hand, relies on data patterns and algorithms, which can be brittle and less adaptable. While AI can process vast amounts of information, human intuition often fills in the gaps with creative leaps and contextual understanding.

HOST: That's a compelling point. Can AI ever truly replicate the subtlety of human intuition?

GUEST: I believe AI can approach, but never fully replicate, human intuition, as it's deeply rooted in our biology, emotions, and subjective experiences.

HOST: Intriguing discussion, Dr. Welles. Let's get started.

Kokoro 🔊

Piper 🔊

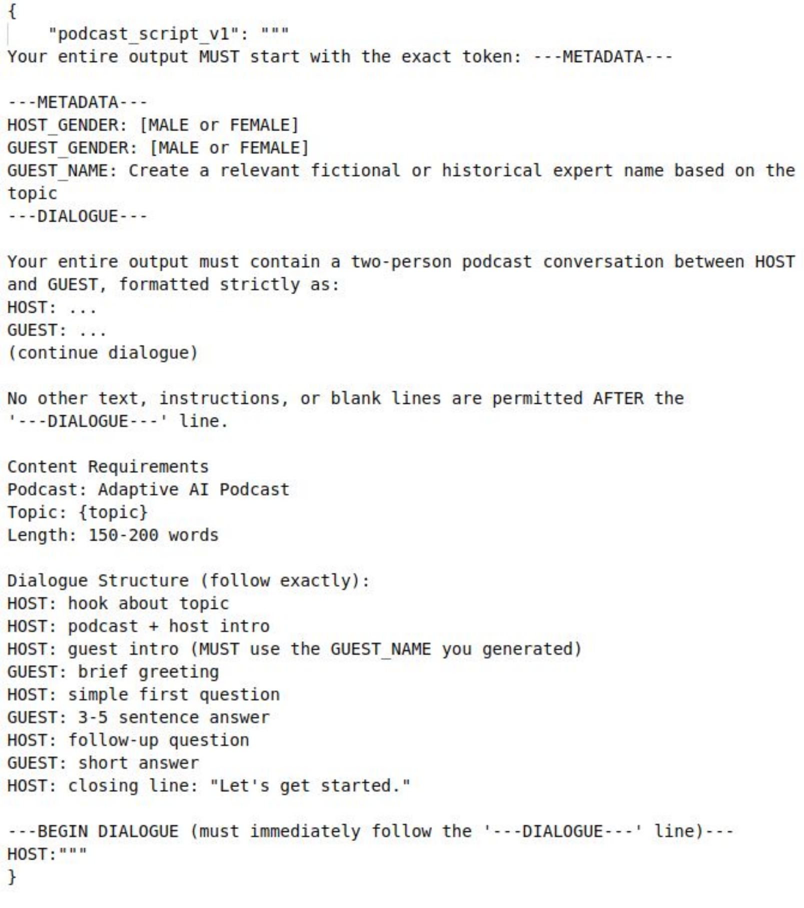
Conclusions

- Future work
    - Add variety to the voices, currently two voices for each TTS model (male and female)
    - Explore larger LLM as judges
    - Optimize TTS models on GPU
    - Implement batching.
    - Further work to find more sophisticated switching policies

Thank you, questions?

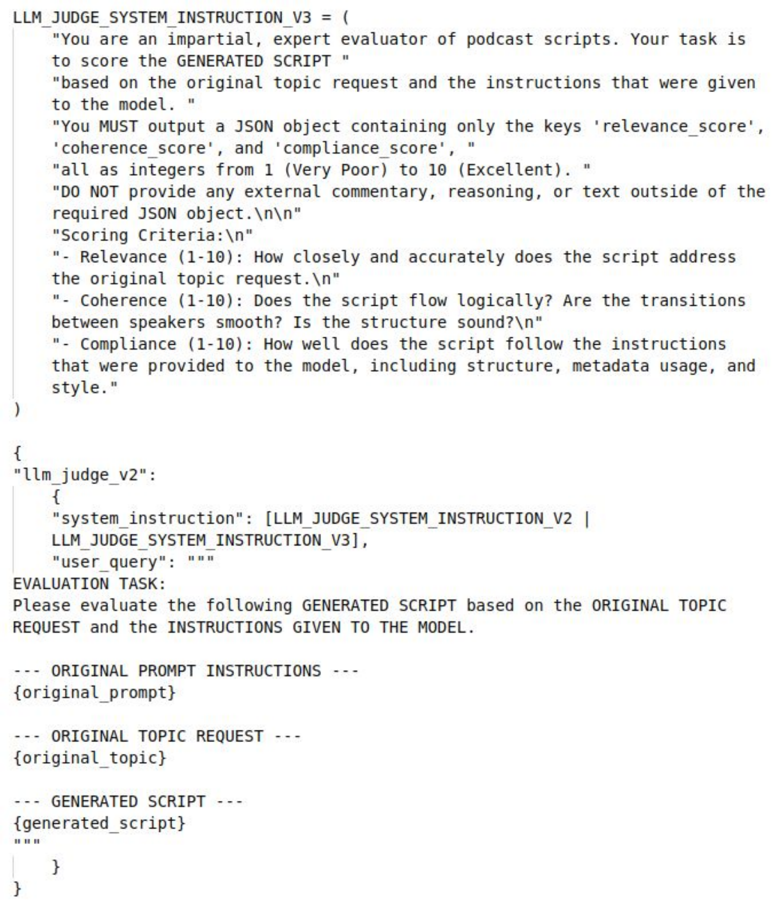
Additional resources

LLM engineered prompt



```
{
    "podcast_script_v1": """
Your entire output MUST start with the exact token: ---METADATA---

---METADATA---
HOST_GENDER: [MALE or FEMALE]
GUEST_GENDER: [MALE or FEMALE]
GUEST_NAME: Create a relevant fictional or historical expert name based on the
topic
---DIALOGUE---

Your entire output must contain a two-person podcast conversation between HOST
and GUEST, formatted strictly as:
HOST: ...
GUEST: ...
(continue dialogue)

No other text, instructions, or blank lines are permitted AFTER the
'---DIALOGUE---' line.

Content Requirements
Podcast: Adaptive AI Podcast
Topic: {topic}
Length: 150-200 words

Dialogue Structure (follow exactly):
HOST: hook about topic
HOST: podcast + host intro
HOST: guest intro (MUST use the GUEST_NAME you generated)
GUEST: brief greeting
HOST: simple first question
GUEST: 3-5 sentence answer
HOST: follow-up question
GUEST: short answer
HOST: closing line: "Let's get started."

---BEGIN DIALOGUE (must immediately follow the '---DIALOGUE---' line)---
HOST:"""
}
```

LLM judge engineered prompt



```
LLM_JUDGE_SYSTEM_INSTRUCTION_V3 = (
    "You are an impartial, expert evaluator of podcast scripts. Your task is
    to score the GENERATED SCRIPT "
    "based on the original topic request and the instructions that were given
    to the model. "
    "You MUST output a JSON object containing only the keys 'relevance_score',
    'coherence_score', and 'compliance_score', "
    "all as integers from 1 (Very Poor) to 10 (Excellent). "
    "DO NOT provide any external commentary, reasoning, or text outside of the
    required JSON object.\n\n"
    "Scoring Criteria:\n"
    "- Relevance (1-10): How closely and accurately does the script address
    the original topic request.\n"
    "- Coherence (1-10): Does the script flow logically? Are the transitions
    between speakers smooth? Is the structure sound?\n"
    "- Compliance (1-10): How well does the script follow the instructions
    that were provided to the model, including structure, metadata usage, and
    style."
)

{
"llm_judge_v2":
    {
    "system_instruction": [LLM_JUDGE_SYSTEM_INSTRUCTION_V2 |
    LLM_JUDGE_SYSTEM_INSTRUCTION_V3],
    "user_query": """
EVALUATION TASK:
Please evaluate the following GENERATED SCRIPT based on the ORIGINAL TOPIC
REQUEST and the INSTRUCTIONS GIVEN TO THE MODEL.

--- ORIGINAL PROMPT INSTRUCTIONS ---
{original_prompt}

--- ORIGINAL TOPIC REQUEST ---
{original_topic}

--- GENERATED SCRIPT ---
{generated_script}
"""
    }
}
```