

Streaming Hindi Conversational Speech System with Real-Time ASR, LLM, and TTS Integration

PARTH PATEL - 202411047

GUIDE: PROF. PRASENJIT MAJUMDER

Outline

- Problem Statement
- Approaches of speech-to-speech modeling
- Motivation
- Methodology & Evaluation
- Demo

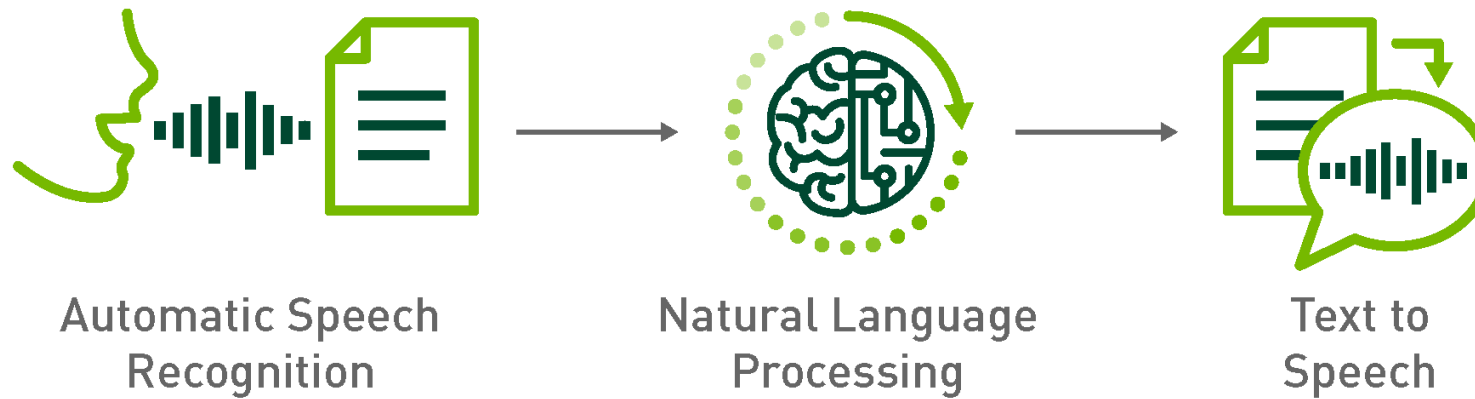
Problem Statement

- Most real-time voice assistants are optimized for high-resource languages like English, neglecting regional languages such as Hindi.
- There is a growing need for millions of native Hindi speakers to engage in conversations with AI systems in their own language.

Various approaches to implement speech-to-speech application

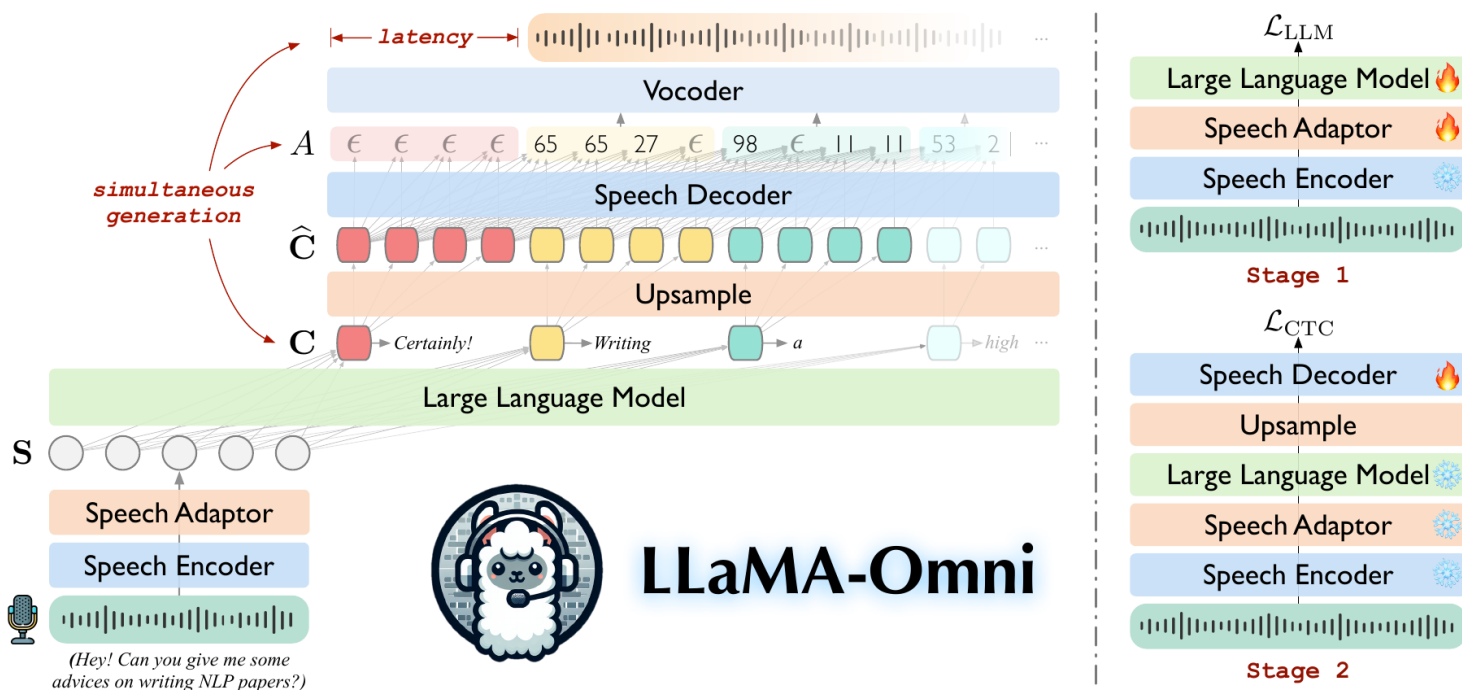
Cascaded approach

- Disadvantages:
 - Latency Issue – Difficult to Deploy for Real-Time Applications



Source: Nvidia

Adaptor based approaches

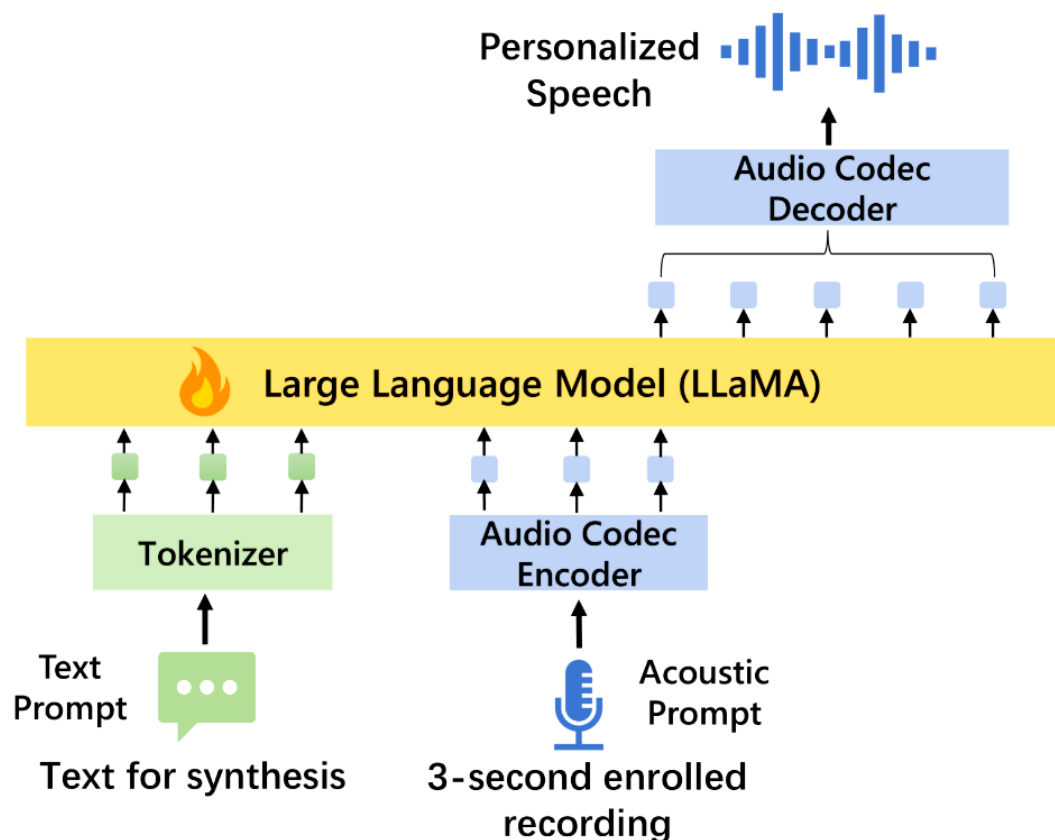


Left: Model architecture of LLaMA-Omni Right: Two stage training process

Fang, Q., Guo, S., Zhou, Y., Ma, Z., Zhang, S., & Feng, Y. (2024). Llama-omni: Seamless speech interaction with large language models. *arXiv preprint arXiv:2409.06666*. (Accepted at ICLR-25)

- Adapters are Lightweight trainable networks inserted into a frozen LLM or speech model to add new functionality.
- Adaptor maps speech representations into the embedding space of the LLM.
- It requires extensive multimodal finetuning, which demands large amounts of speech-text data and high computational resources.
- Fine-tuning harms the base LLM's original reasoning and expressiveness.
- This speech adaptation often conditions on hidden states of LLM, which makes it LLM-dependent requiring re-adaptation for each base LLM.

Acoustic token-based approaches



- Idea is to extend the vocabulary by incorporating discrete acoustic tokens alongside text tokens.
- The LLM is fine-tuned to predict these audio tokens, which are then passed to an audio codec decoder to synthesize speech.
- An enhanced version of this approach is capable of generating both text and audio tokens simultaneously, enabling unified multimodal output.

Adding acoustic tokens into vocabulary of LLM

Hao, H., Zhou, L., Liu, S., Li, J., Hu, S., Wang, R., & Wei, F. (2025, April). Boosting large language model for speech synthesis: An empirical study. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 1-5). IEEE.

Streaming Text-to-Speech approach

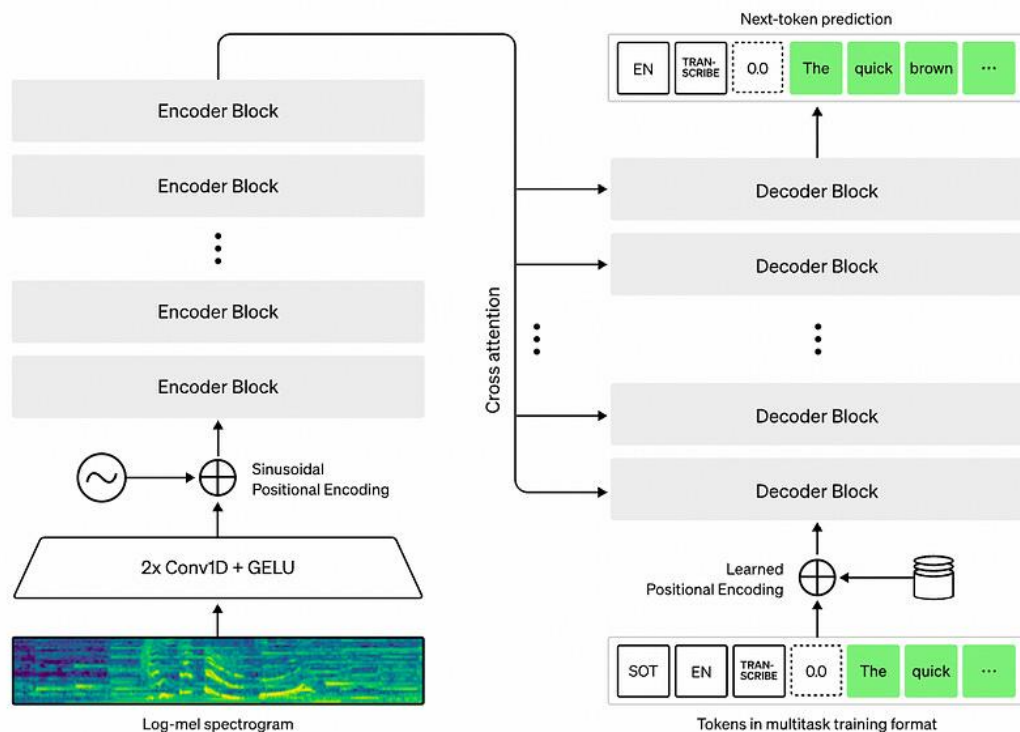
- Traditional TTS systems wait for the LLM to generate the full response before synthesizing audio, which introduces latency.
- Also, they often require fine-tuning the LLM, which can degrade its performance.
- Streaming TTS decouples speech synthesis from the LLM, allowing both to operate in parallel.
- This approach enables the use of any pretrained LLM without modification, ensuring flexibility and preserving response quality.

Motivation to use streaming TTS

- For Hindi speech modeling, to use audio token-based approach, we need very large Hindi speech to speech conversational question answering dataset.
- But as of now there is none dataset of this kind is available.
- Adaptor based approaches requires very high computational resources to finetune model in 2 or 3 different training stages.
- Also, this finetuning is LLM specific and does not work for another LLM.
- So, In order to overcome these issues we decided to use streaming TTS approach for Hindi speech generation.
- And to use whisper model to convert speech to text.

Methodology & Evaluation

Automatic Speech Recognition (ASR)



- Used Whisper model[1] for speech recognition.
- Used Silero-VAD[2] for voice activity detection.
- VAD used to detect Voice and Noise
- VAD - 120M parameters

Model (Whisper)	Base	Small	Medium	Large
Parameter Size	74 M	244 M	769 M	1550 M
VRAM	426 MB	778 MB	1802 MB	3306 MB

Whisper-Openai

[1] Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., & Sutskever, I. (2023, July). Robust speech recognition via large-scale weak supervision. In *International conference on machine learning* (pp. 28492-28518). PMLR.

[2] Team, S. (2024). Silero VAD: pre-trained enterprise-grade Voice Activity Detector (VAD), Number Detector and Language Classifier. *GitHub Repository*. Retrieved from <https://github.com/snakers4/silero-vad>

ASR Evaluation

- We have evaluated ASR transcription by 10 various Hindi prompts on Whisper model.
- $Word\ Error\ Rate = \frac{S + D + I}{N}$
- S = Number of Substitutions
- D = Number of Deletions
- I = Number of Insertions
- N = Number of words in the reference transcription

Model	Base	Small	Medium	Large
Word Error Rate	-	17.9%	11.94%	11.94%
Latency (Seconds)	0.11	0.20	0.38	0.53

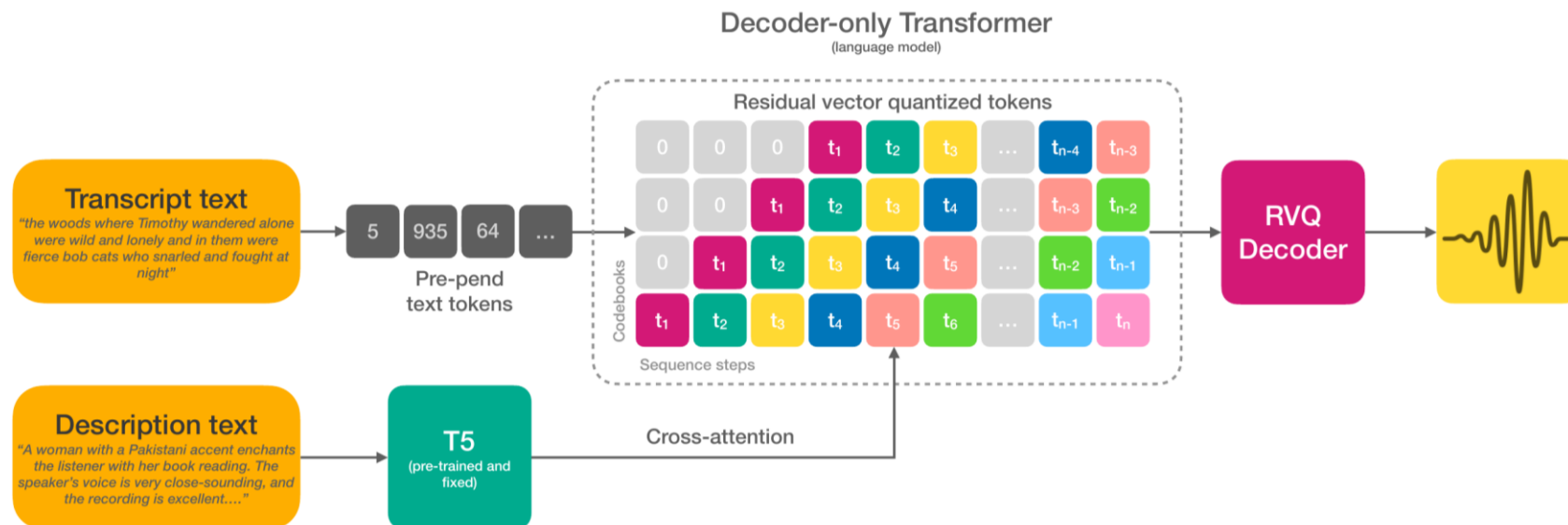
LLM Inference

- We have used “Meta-Llama-3.2-3B-it” & “Meta-Llama-3.2-1B-it” model as a base LLM.
- We have prompted each LLM to evaluate latency, further due to memory constraint we decided to use cloud LLM for our application.
- System prompt: You are a helpful voice AI assistant. Keep your answer short and don't use abbreviation and special tokens. Give your reply in hindi language only.

LLM	3B Local	3B Cloud	1B Local	1B Cloud
Time to first token	112 ms	680 ms	60 ms	682 ms
Token per second	64 tps	81 tps	148 tps	119 tps
VRAM	9104 MB	-	3816 MB	-

Streaming TTS

- We have converted pre-trained “ai4bharat/indic-parler-tts”[1] into a streaming TTS by chunking output.
- It has 938 M parameters which uses 4586 MB of VRAM.
- It is a fine-tuned version of Indic-Parler-TTS-Pretrained(Finetuned on Parler-TTS-Mini with 8,385 hours of multilingual Indic and English dataset), trained on a 1,806 hours multilingual Indic and English dataset.
- It can speak in 21 languages, namely Assamese, Bengali, Bodo, Dogri, English, Gujarati, Hindi, Kannada, Konkani, Maithili, Malayalam, Manipuri, Marathi, Nepali, Odia, Sanskrit, Santali, Sindhi, Tamil, Telugu, and Urdu.



Model Architecture

[1] Lacombe, Y., Sankar, A., Thomas, S., Varadhan, P. S., Gandhi, S., & Khapra, M. (2024). Indic Parler-TTS [Hugging Face repository]. Hugging Face. <https://huggingface.co/ai4bharat/indic-parler-tts>

TTS Evaluation

- We have used 2 streaming pre-trained “ai4bharat/indic-parler-tts” in parallel to synthesize speech from text.
- Which uses 9172 MB of VRAM.
- It takes input from 2 variable length FIFO queues.
- Initial chunk size decides latency. After every toggling chunk size get doubled.

Initial Chunk Size (Char)	Latency (Seconds)
50	4.53
40	3.09
30	2.90
20	2.09
10	1.73

Speech quality tradeoff

- We used the Mean Opinion Score (MOS) to evaluate the naturalness or quality of speech.
- The MOS score is determined by a encoder(CNN) neural network[1] trained on human-rated naturalness scores ranging from 1 to 5.
- We have evaluated it on 10 different recordings of each size of chunk.

Initial Chunk Size (Char)	Mean MOS Score
50	3.03
40	3.08
30	3.01
20	2.89
10	2.85

[1] Baba, K., Nakata, W., Saito, Y., & Saruwatari, H. (2024). The T05 System for The VoiceMOS Challenge 2024: Transfer Learning from Deep Image Classifier to Naturalness MOS Prediction of High-Quality Synthetic Speech. *IEEE Spoken Language Technology Workshop (SLT)*.

Combining all components

- ASR : Whisper-Medium (Local)
- LLM : Meta-Llama-3.2-3B-it (Cloud)
- TTS : Indic-parler-tts (Local)
- Total Size (Vram) : 10974 MB

Initial Chunk Size (Char)	Overall average latency (Second)
10	3.73
20	6

Peer comparison

Model	Base LLM	MOS (1-5)	Latency (s)
SpeechGPT[1]	Llama 2 13B	3.86	4
Mini-omni[2]	Qwen2 0.5B	3.24	0.35
Llama-omni[3]	Llama 3.1 8B	3.32	0.22
Moshi[4]	Helium 7B	3.92	0.32
GLM-4-Voice[5]	GLM-4 9B	3.97	2.5
Freeze-omni[6]	Qwen2 7B	4.38	0.34
MiniCPM-o 2.6[7]	Qwen2.5 7B	3.87	1.2
LLMVoX[8]	Llama 3.1 8B	4.05	0.47
Whisper+LLM+XTTS[9]	Llama 3.1 8B	4.23	4.2
Ours (Hindi)	Llama 3.2 3B	2.85	3.73

[1] Zhang, D., Li, S., Zhang, X., Zhan, J., Wang, P., Zhou, Y., & Qiu, X. (2023). Speechgpt: Empowering large language models with intrinsic cross-modal conversational abilities. arXiv preprint arXiv:2305.11000.

[2] Xie, Z., & Wu, C. (2024). Mini-omni: Language models can hear, talk while thinking in streaming. arXiv preprint arXiv:2408.16725.

[3] Fang, Q., Guo, S., Zhou, Y., Ma, Z., Zhang, S., & Feng, Y. (2024). Llama-omni: Seamless speech interaction with large language models. arXiv preprint arXiv:2409.06666.

[4] Défossez, A., Mazaré, L., Orsini, M., Royer, A., Pérez, P., Jégou, H., ... & Zeghidour, N. (2024). Moshi: a speech-text foundation model for real-time dialogue. arXiv preprint arXiv:2410.00037.

[5] Zeng, A., Du, Z., Liu, M., Wang, K., Jiang, S., Zhao, L., ... & Tang, J. (2024). Glm-4-voice: Towards intelligent and human-like end-to-end spoken chatbot. arXiv preprint arXiv:2412.02612.



[6] Wang, X., Li, Y., Fu, C., Shen, Y., Xie, L., Li, K., ... & Ma, L. (2024). Freeze-omni: A smart and low latency speech-to-speech dialogue model with frozen llm. arXiv preprint arXiv:2411.00774.

[7] Yao, Y., Yu, T., Zhang, A., Wang, C., Cui, J., Zhu, H., Cai, T., Li, H., Zhao, W., He, Z., & others. (2024). MiniCPM-V: A GPT-4V level MLLM on your phone. arXiv preprint arXiv:2408.01800. arXiv preprint arXiv: 2408.01800










[8] Shikhar, S., Kurpath, M. I., Mullappilly, S. S., Lahoud, J., Khan, F., Anwer, R. M., ... & Cholakkal, H. (2025). LLMVoX: Autoregressive Streaming Text-to-Speech Model for Any LLM. arXiv preprint arXiv:2503.04724.





[9] Casanova, E., Davis, K., Gölge, E., Gökner, G., Gulea, I., Hart, L., ... & Weber, J. (2024). Xtts: a massively multilingual zero-shot text-to-speech model. arXiv preprint arXiv:2406.04904.

Demo - Initial chunk size 10

 jupyter full_model evaluation Last Checkpoint: 1 hour ago 








File Edit View Run Kernel Settings Help Trusted

 +        Code 

JupyterLab   Python 3 (ipykernel)  


```
"dropout": 0.1,
"eos_token_id": 1024,
"ffn_dim": 4096,
"hidden_size": 1024,
"initializer_factor": 0.02,
"is_decoder": true,
"layerdrop": 0.0,
"max_position_embeddings": 4096,
"model_type": "parler_tts_decoder",
"num_attention_heads": 16,
"num_codebooks": 9,
"num_cross_attention_key_value_heads": 16,
"num_hidden_layers": 24,
"num_key_value_heads": 16,
"pad_token_id": 1024,
"rope_embeddings": false,
"rope_theta": 10000.0,
"scale_embedding": false,
"tie_word_embeddings": false,
"torch_dtype": "float32",
"transformers_version": "4.51.3",
"use_cache": true,
"use_fused_lm_heads": true,
"vocab_size": 1088
}
```

Wait until it says 'speak now'











 []:      


[]:

Demo - Initial chunk size 20

 jupyter full_model evaluation Last Checkpoint: 3 hours ago

File Edit View Run Kernel Settings Help

JupyterLab  Python 3 (ip

```
"ParlerTTSForCausalLM"
],
"attention_dropout": 0.0,
"bos_token_id": 1025,
"codebook_weights": null,
"cross_attention_implementation_strategy": null,
"delay_strategy": "delay",
"dropout": 0.1,
"eos_token_id": 1024,
"ffn_dim": 4096,
"hidden_size": 1024,
"initializer_factor": 0.02,
"is_decoder": true,
"layerdrop": 0.0,
"max_position_embeddings": 4096,
"model_type": "parler_tts_decoder",
"num_attention_heads": 16,
"num_codebooks": 9,
"num_cross_attention_key_value_heads": 16,
"num_hidden_layers": 24,
"num_key_value_heads": 16,
"pad_token_id": 1024,
"rope_embeddings": false,
"rope_theta": 10000.0,
"scale_embedding": false,
"tie_word_embeddings": false,
"torch_dtype": "float32",
"transformers_version": "4.51.3",
"use_cache": true,
"use_fused_lm_heads": true,
"vocab_size": 1088
}

Wait until it says 'speak now'
Model medium completed transcription in 0.29 seconds
आपका स्वागत है! कैसे हैं आप? हैं आप?WARNING:parler_tts.modeling_parler_tts:`prompt_attention_mask` is specified but `attention_mask` is not. A full `attention_mask` will be created. Make sure this is the intended behaviour.

Latency: 1.86s
```

Fullscreen Pro

Fullscreen Pro

Start Stream

Start Recording

Start Replay

Start Virtual

Exit

Observations

- There is a trade off between latency and speech quality, as latency depends on initial chunk size.
- By using powerful GPU which has high cuda cores and vram, we can further decrease latency by using LLM locally. It can reduce latency by 1 to 2 seconds.
- Our initial attempts to train a 30M TTS model from scratch revealed that a smaller model and limited dataset are insufficient for effectively synthesizing Hindi speech.

Thank You