Ubaid
Ghante

Task Results

INDEX

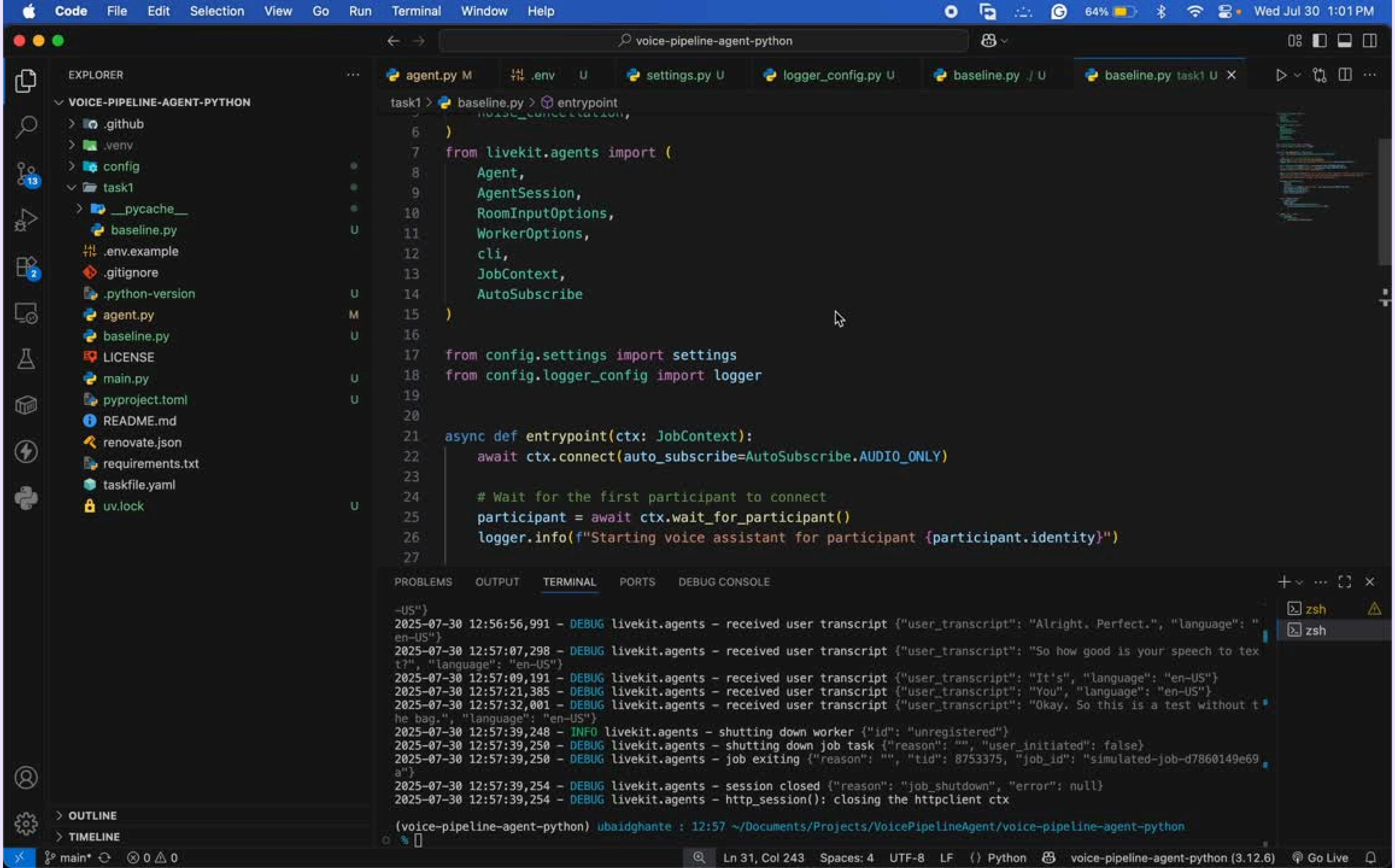
1

Setup & Baseline
Background

Forked to https://github.com/Ubaid-Ghante/voice-pipeline-agent-python

Simple implementation using deepgram, cartesia and openai.
Created accounts in all the above services and got the API keys.

The example code give in Assignment seems for the older version.
Next slide contains a simple demo of the system working.


Code File Edit Selection View Go Run Terminal Window Help
64% Wed Jul 30 1:01 PM
voice-pipeline-agent-python
agent.py M
.env U
settings.py U
logger_config.py U
baseline.py ./ U
baseline.py task1 U
EXPLORER
VOICE-PIPELINE-AGENT-PYTHON
.github
.venv
config
task1
__pycache__
baseline.py U
.env.example
.gitignore
.python-version U
agent.py M
baseline.py U
LICENSE
main.py U
pyproject.toml U
README.md
renovate.json
requirements.txt
taskfile.yaml
uv.lock U
OUTLINE
TIMELINE
task1 > baseline.py > entrypoint
)
from livekit.agents import (
Agent,
AgentSession,
RoomInputOptions,
WorkerOptions,
cli,
JobContext,
AutoSubscribe
)
from config.settings import settings
from config.logger_config import logger
async def entrypoint(ctx: JobContext):
await ctx.connect(auto_subscribe=AutoSubscribe.AUDIO_ONLY)
# Wait for the first participant to connect
participant = await ctx.wait_for_participant()
logger.info(f"Starting voice assistant for participant {participant.identity}")
PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE
-US"}
2025-07-30 12:56:56,991 - DEBUG livekit.agents - received user transcript {"user_transcript": "Alright. Perfect.", "language": "en-US"}
2025-07-30 12:57:07,298 - DEBUG livekit.agents - received user transcript {"user_transcript": "So how good is your speech to text?", "language": "en-US"}
2025-07-30 12:57:09,191 - DEBUG livekit.agents - received user transcript {"user_transcript": "It's", "language": "en-US"}
2025-07-30 12:57:21,385 - DEBUG livekit.agents - received user transcript {"user_transcript": "You", "language": "en-US"}
2025-07-30 12:57:32,001 - DEBUG livekit.agents - received user transcript {"user_transcript": "Okay. So this is a test without the bag.", "language": "en-US"}
2025-07-30 12:57:39,248 - INFO livekit.agents - shutting down worker {"id": "unregistered"}
2025-07-30 12:57:39,250 - DEBUG livekit.agents - shutting down job task {"reason": "", "user_initiated": false}
2025-07-30 12:57:39,250 - DEBUG livekit.agents - job exiting {"reason": "", "tid": 8753375, "job_id": "simulated-job-d7860149e69a"}
2025-07-30 12:57:39,254 - DEBUG livekit.agents - session closed {"reason": "job_shutdown", "error": null}
2025-07-30 12:57:39,254 - DEBUG livekit.agents - http_session(): closing the httpclient ctx
(voice-pipeline-agent-python) ubaidghante : 12:57 ~/Documents/Projects/VoicePipelineAgent/voice-pipeline-agent-python
zsh
zsh
main* 0 0
Ln 31, Col 243 Spaces: 4 UTF-8 LF Python voice-pipeline-agent-python (3.12.6) Go Live

2

VAD / Endpointing Parameter Tuning

Min 0.2s and **Max** 3s -

Total latency :
2.0999720122199506 seconds

Min 2.0s and **Max** 6s -

Total latency :
3.34827495040372 seconds

Observation:

- Mainly added idle time if both min and max kept too high
- Starts interrupting if kept too low. High false triggers during natural pauses.

Ubaid
Ghante

3

Turn Detector Plugin (Model-based)

Observation:

- Makes it way easier to talk without any false trigger.
- Allows **long pause** where it is needed.
- If eou probabilty threshold in not specified it takes from the pre configured hf **"livekit/turn-detector"** downloaded file.
- It waits for around 6 to 7 seconds after taking a pause on I think..

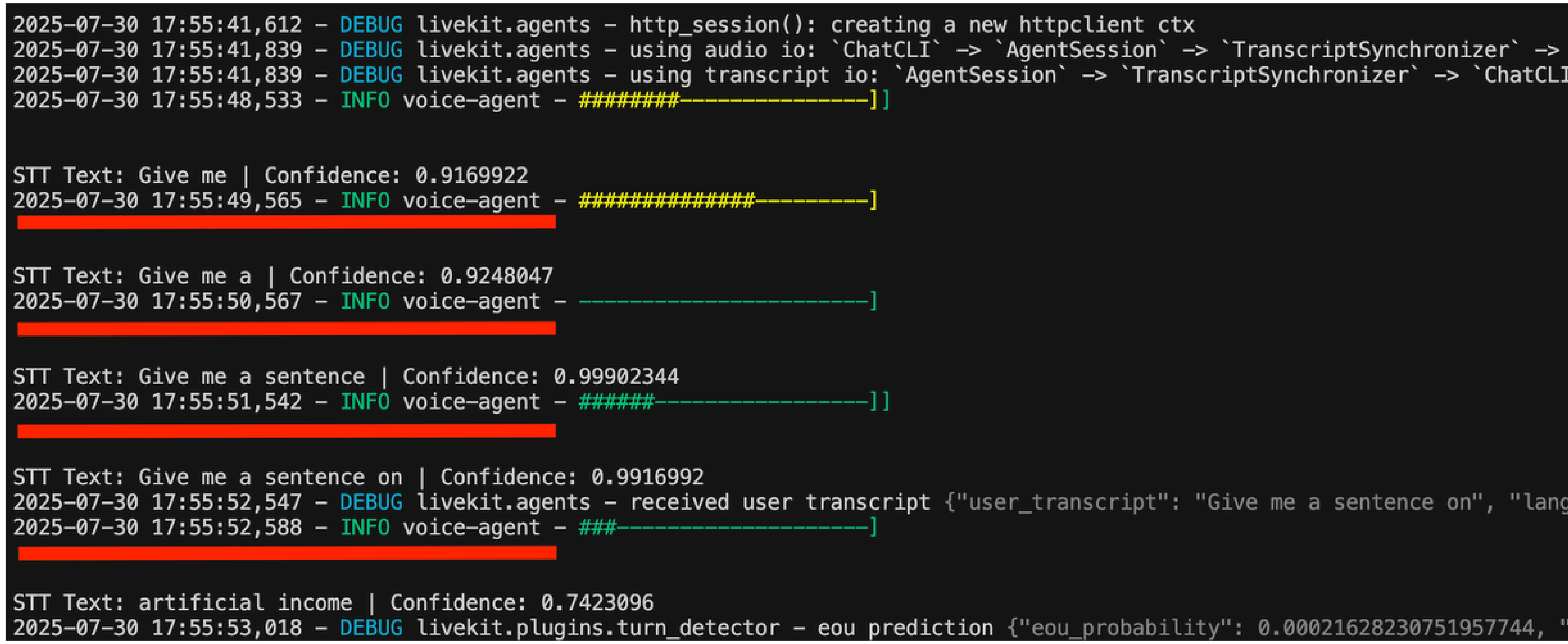
```
DEBUG livekit.agents - received user transcript {"user_transcript": "I think", "language": "en-US"}
2025-07-30 16:29:19,352 - DEBUG livekit.plugins.turn_detector - eou prediction {"eou_probability": 0.001277949777431786, "input"
me a sentence about artificial intelligence<|im_end|>\n<|im_start|>assistant\nhowdy artificial intelli
and decision-making in many fields<|im_end|>\n<|im_start|>user\ni think", "duration": 0.094}
2025-07-30 16:29:26,680 - INFO voice-agent - TTS Metrics: type='tts_metrics' label='livekit.plugins.cartesia.tts.TTS' request_id
1753873166.6802402 ttfb=0.2818183330819011 duration=1.0776705420576036 audio_duration=2.46129166666666
64 cancelled=False characters_count=31 streamed=True segment_id='584bfa35c9c0' speech_id='speech_7973bfdcad88'
```

4

Streaming STT & Partial Hypotheses

Observations

- Added stt_node overriding to log results
- Using preemptive genration reduces latency by a little.
- Experimented with basic examples. Major problem seems to be STT, WER is very high.
- LLM Hallucination is very less since most of them are generic test.



5

SSML-Driven TTS Enhancements

Observation:

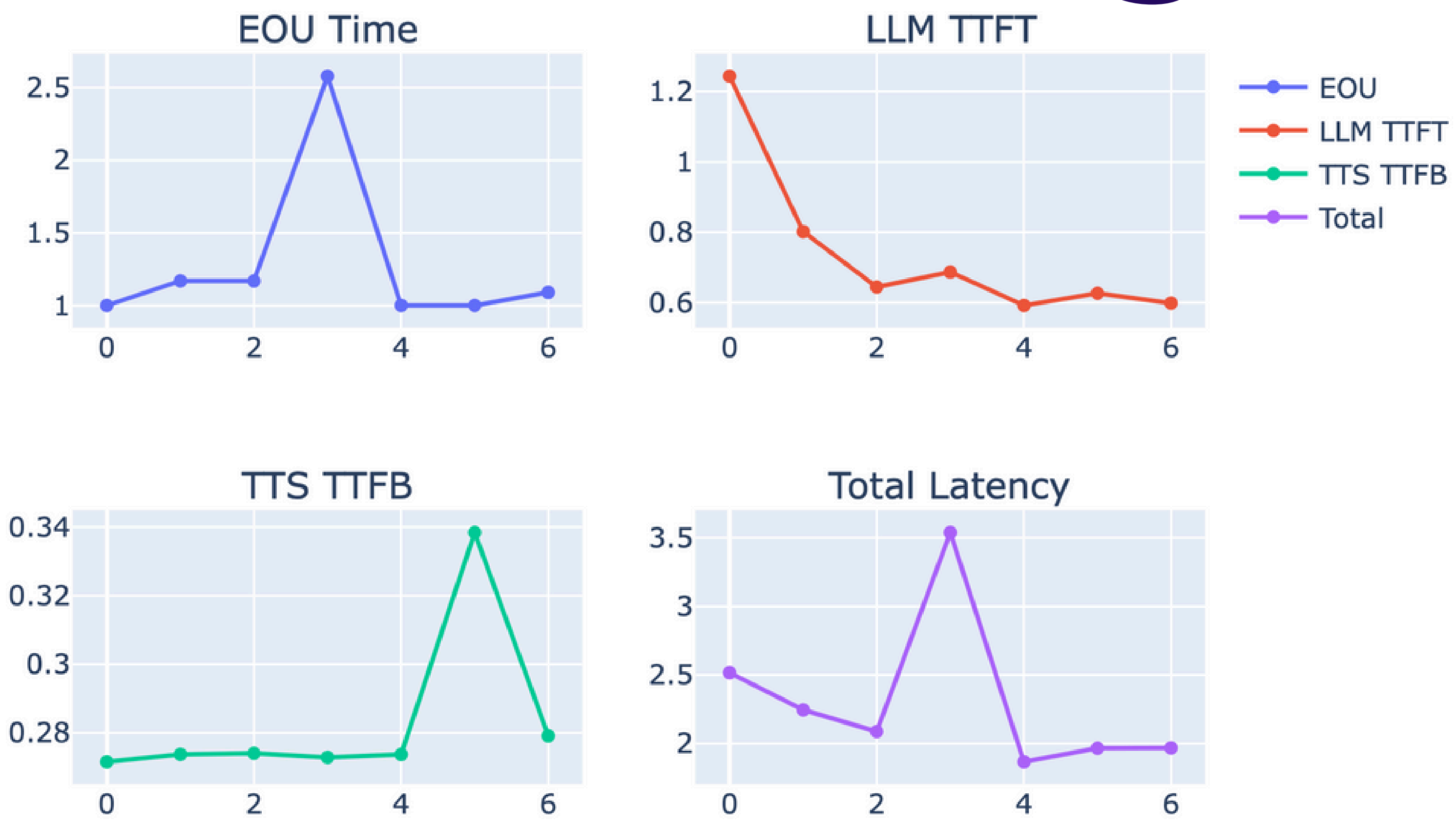
- prosody works fine and allows pitch changes
- Spelling out words
- Having different pronunciations for words like ngnix
- emphasis tag was not being responded too.

MOS Score - 3

- Rushes most part of the sentence.
- Puntuation pauses are not that great.

6

Latency & Quality Benchmarking



Baseline
MOS - 2

Advanced
MOS - 3

6

Latency & Quality Benchmarking

Observation:

- Baseline pipeline has fast responses from LLMs since the prompts are small and simple, but this reduces the accuracy of the response.
- Advance pipeline has a better prompt and does not generate stuff that cannot be spelt out, like having * for bold.
- TTS takes a little more time in Advance pipeline to do all the preprocessing compared to the baseline one.
- The overall latency is lower in Advance pipeline due to preemptive generation.
- Baseline model interrupts a lot while taking a pause.
- Since models are the same in both pipelines, the WER looks the same.

Ubaid
Ghante

I-Stem
2025

</> Code Repo

Thank You.

Get In Touch With Us

+91 9284876115

ughante@gmail.com

LinkedIn