

Project Introduction

- The primary goal is to process raw audio inputs and generate sentiment predictions through sequential speech-to-text transcription and sentiment classification.
- It aims to establish a modular and scalable architecture capable of integrating multiple ASR and sentiment models.
- The objective is to benchmark these model combinations based on accuracy, latency, throughput, and overall efficiency.
- Ultimately, it seeks to evolve into an operational and deployable system suitable for real-time audio sentiment analysis.

Group Members

- Ziyang Jia (Pipeline Composition, Container-based Service)
- Srikar Alla (Pipeline Composition, Container-based Service)
- Souvik Bag (Evaluation, Report)

Settings

Models Included

1. Audio to Text
 - AventIQ-AI/whisper-audio-to-text
 - Base Architecture: OpenAI Whisper
 - Dataset / Fine-Tuning: Mozilla Common Voice 13.0 dataset
 - Task / Use-Case: Speech-to-Text (Automatic Speech Recognition, ASR)
 - Parameter Size: 72M
1. Text to Sentiment
 - distilbert-base-uncased-finetuned-sst-2-english (Small)
 - Base Architecture: DistilBERT-base-uncased
 - Dataset / Fine-Tuning: Stanford Sentiment Treebank (SST-2)
 - Task / Use-Case: Sentiment Analysis
 - Parameter Size: 66M
 - siebert/sentiment-roberta-large-english (Large)
 - Base Architecture: RoBERTa-Large
 - Dataset / Fine-Tuning: 15 mixed English sentiment datasets
 - Task / Use-Case: Sentiment Analysis
 - Parameter Size: 355M
 - cardiffnlp/twitter-roberta-base-sentiment-latest (Medium)
 - Base Architecture: RoBERTa-Base
 - Dataset / Fine-Tuning: 124 M tweets (TweetEval benchmark)
 - Task / Use-Case: Sentiment Analysis
 - Parameter Size: 125M

Model Variant

1. Quantitatively Evaluated
 - fp32 - audio to text models and text to sentiment models
 - fp16 - text to sentiment models
1. Provided in Web Service (will be quantitatively evaluated in Milestone 3)
 - fp32 - audio to text models and text to sentiment models
 - int8 - text to sentiment models

Model Storage

- MinIO
 - An open-source, high-performance object storage server compatible with Amazon S3 APIs.
 - Usage in this pipeline: Store our models in ONNX

Temporal File Storage

- Redis
 - A high-speed, in-memory key-value store that can act as a cache, database, or message broker.
 - Usage in this pipeline: Cache frequent inference results to reduce model load.

Web Server

- FastAPI
 - A lightweight, asynchronous web framework built on Starlette (for async I/O) and Pydantic (for data validation).
 - Usage in this pipeline: deploy docker containers as web service.

Flexible Container-Based Web Service:

- We kept both audio-to-text and text-to-sentiment models as flexible options when we create a container
- We assigned different ports to different model combinations
- Example:
 - input: .wav file
 - output:
 - transcribed text
 - sentiment classification
 - inference time

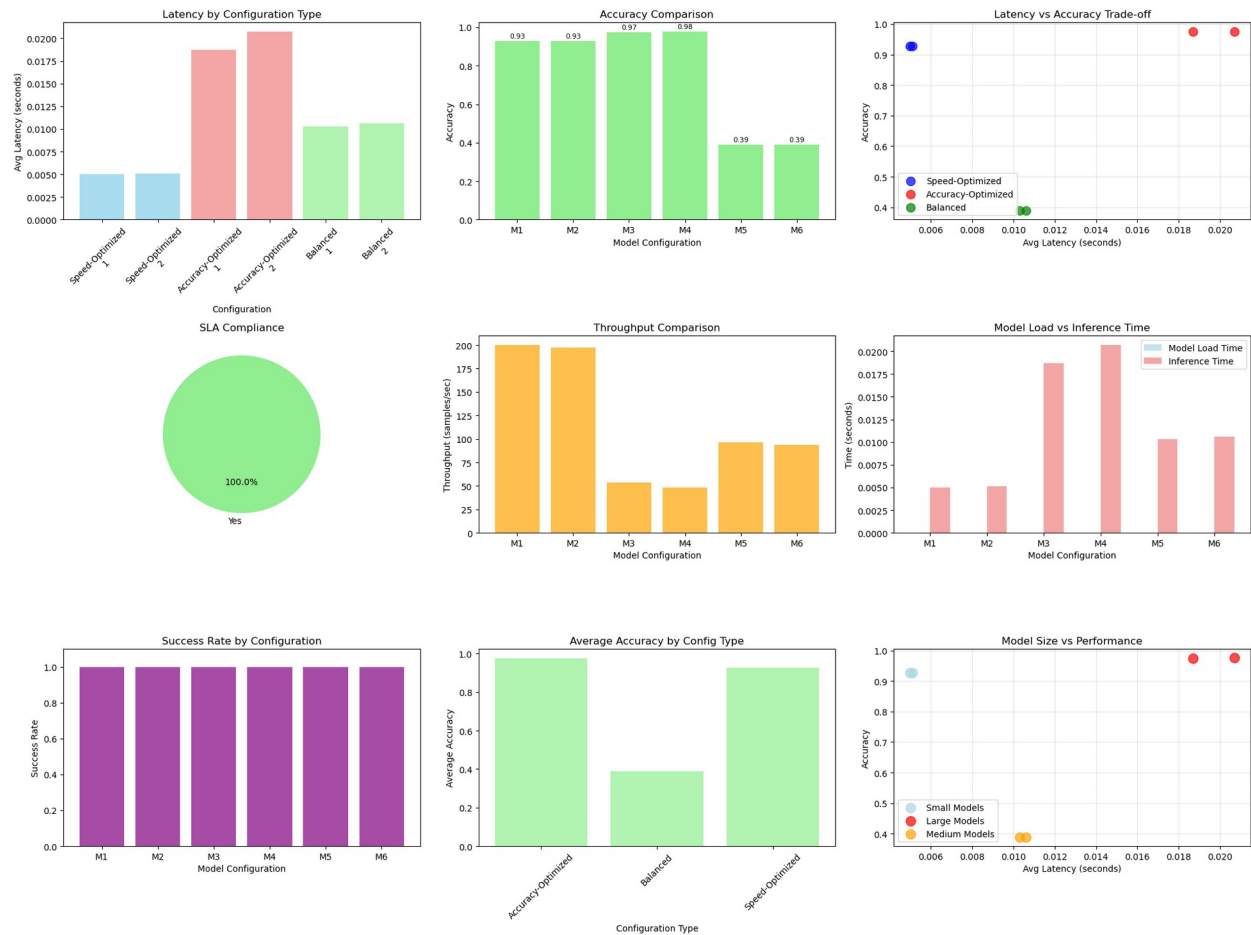
```
Windows PowerShell
(whisper_export) PS D:\Mizzou\25fall\ee\project\ee2eml-audio\ms_2> docker run -d -p 8002:8000
--e A2T_MODEL_PATH=/models/a2t_AventIQ-AI_whisper-audio-to-text_fp32"
--e T2S_MODEL_PATH=/models/t2s_distilbert-base-uncased-finetuned-sst-2-english_fp32.onnx"
--v $PWD:/app:/app
--v "${PWD}/models:/models"
audio-sentiment
d9042b02240a979ed0b68e127cbf2a1ee6edb9b5bead1458387571cf1b962122
(whisper_export) PS D:\Mizzou\25fall\ee\project\ee2eml-audio\ms_2> docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS
NAMES
d9042b02240a   audio-sentiment "uvicorn main:app --..." 4 seconds ago  Up 4 seconds  0.0.0.0:8002->8000/tcp, [::]:8002->8000/tcp
intelligent_zhukovsky
(whisper_export) PS D:\Mizzou\25fall\ee\project\ee2eml-audio\ms_2> $FilePath = "D:\Mizzou\25fall\ee\project\ee2eml-audio\data\audio\audio_000000.wav"
(whisper_export) PS D:\Mizzou\25fall\ee\project\ee2eml-audio\ms_2> $FileContent = [System.IO.File]::ReadAllBytes($FilePath)
(whisper_export) PS D:\Mizzou\25fall\ee\project\ee2eml-audio\ms_2> $FileName = [System.IO.Path]::GetFileName($FilePath)
(whisper_export) PS D:\Mizzou\25fall\ee\project\ee2eml-audio\ms_2> $boundary = "----WebKitFormBoundary${[System.Guid]::NewGuid()}.ToString()"
(whisper_export) PS D:\Mizzou\25fall\ee\project\ee2eml-audio\ms_2>
(whisper_export) PS D:\Mizzou\25fall\ee\project\ee2eml-audio\ms_2> $bodyLines = @(
--$boundary
"Content-Disposition: form-data; name='audio'; filename='$FileName'"
"Content-Type: audio/wav"
""
[System.Text.Encoding]::UTF8.GetString($FileContent)
--$boundary--
) -join "`n"
(whisper_export) PS D:\Mizzou\25fall\ee\project\ee2eml-audio\ms_2>
(whisper_export) PS D:\Mizzou\25fall\ee\project\ee2eml-audio\ms_2> Invoke-RestMethod -Uri "http://localhost:8002/predict" `
-Method POST `
-ContentType "multipart/form-data; boundary=$boundary" `
-Body $bodyLines

transcript                                     sentiment
-----
Merely, he is a man of the same name as the king of the world. @{label=POSITIVE; confidence=0.7689058184623718}

(whisper_export) PS D:\Mizzou\25fall\ee\project\ee2eml-audio\ms_2> Measure-Command {
$FilePath = "D:\Mizzou\25fall\ee\project\ee2eml-audio\data\audio\audio_000000.wav"
$FileContent = [System.IO.File]::ReadAllBytes($FilePath)
$FileName = [System.IO.Path]::GetFileName($FilePath)
$boundary = "----WebKitFormBoundary${[System.Guid]::NewGuid()}.ToString()"
$bodyLines = @(
--$boundary
"Content-Disposition: form-data; name='audio'; filename='$FileName'"
"Content-Type: audio/wav"
""
[System.Text.Encoding]::UTF8.GetString($FileContent)
--$boundary--
) -join "`n"
Invoke-RestMethod -Uri "http://localhost:8002/predict" `
-Method POST `
-ContentType "multipart/form-data; boundary=$boundary" `
-Body $bodyLines
}

Days           : 0
Hours          : 0
Minutes        : 0
Seconds        : 0
Milliseconds   : 821
Ticks          : 8211151
TotalDays      : 9.50364699074074E-06
TotalHours     : 0.000228087527777778
TotalMinutes   : 0.0136852516666667
```

Evaluation Plots



In the plots above, we used the same audio-to-text model(AventIQ-AI/whisper-audio-to-text fp32) and different vairants of text-to-sentiment models:

- M1: distilbert-base-uncased-finetuned-sst-2-english (fp32)
- M2: distilbert-base-uncased-finetuned-sst-2-english (fp16)
- M3: siebert/sentiment-roberta-large-english (fp32)
- M4: siebert/sentiment-roberta-large-english (fp16)
- M5: cardiffnlp/twitter-roberta-base-sentiment-latest (fp32)
- M6: cardiffnlp/twitter-roberta-base-sentiment-latest (fp16)

Variant	Scale	Avg Latency (s)	Accuracy	Throughput (samples/sec)	Relative Cost (\approx Compute Load)	Notes
M1 (Speed-Optimized 1)	Small	0.005 s	0.93	200	\$ Low	Fastest; ideal for real-time use; small model size
M2 (Speed-Optimized 2)	Small	0.005 s	0.93	195	\$ Low	Similar to M1; nearly identical accuracy & latency
M3 (Accuracy-Optimized 1)	Large	0.018 s	0.97	55	\$ \$ \$ High	High-accuracy but slower; high compute cost
M4 (Accuracy-Optimized 2)	Large	0.021 s	0.98	50	\$ \$ \$ High	Highest accuracy; highest latency
M5 (Balanced 1)	Medium	0.010 s	0.39	100	\$ \$ Medium	Trade-off config; moderate speed, low accuracy
M6 (Balanced 2)	Medium	0.010 s	0.39	95	\$ \$ Medium	Similar to M5; balanced cost/speed but weak accuracy

Empirical Conclusions:

- the differences on precision (fp32 vs fp16)
 - doesn't impact too much in terms of accuracy and throughput
 - impacts more on inference latency
- the model structure impacts more on temporal efficiency, e.g. given acc=0.93 and 0.97 for M1(fp16) and M3(fp16), we found that
 - the latency of M1 is only around 1/4 of M3
 - the throughput of M3 is only 1/4 of M1
 - this observation holds for other comparison pairs as well
- the model structure impacts more on accuracy as well, e.g. given parameter_size=66M and 125M for M1(fp16) and M5(fp16), the temporal efficiency pattern still holds but we found:
 - the accuracy of M1 is more than twice of M5's accuracy
- some conclusions in model selection and optimization:
 - some model structure+pretrained dataset settings inherently get better accuracies in sentiment classification
 - precision impacts the accuracy but the difference is trivial in this task and in real-world applications, fp16 is more promising than the original fp32 models (in terms of temporal efficiency)

- inspirations for our future plan in milestone 3: we'll
 - push the temporal efficiency further to explore the capability of int8 models
 - try more flexible strategies for model choosing given different input (will consider some features like size of the input audio)
 - provide an interactive webpage if time permits