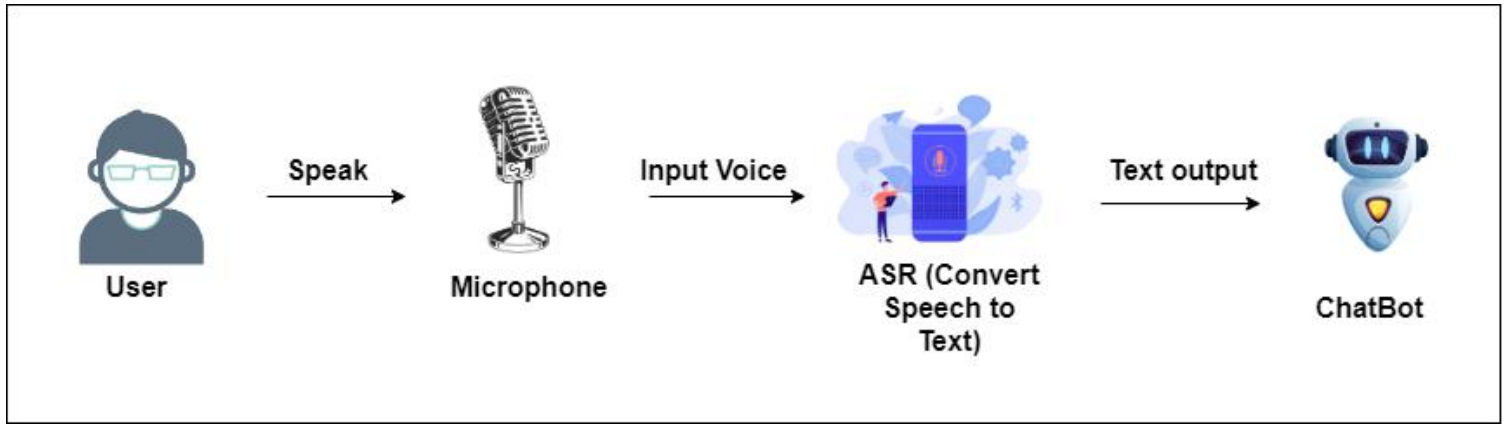
IT20237554 | RATHNAWEERA R.P.W.G

B.Sc. (Hons) Degree in Information Technology Specialized in Data Science

Specific and Sub Objectives

Main Objective

Development of an Automatic Speech Recognition system for feeding quires to Chatbot



Specific and Sub Objectives

- Sub Objective
  - Dataset creation
    - Developed custom data scrappers to gather data for the recommendation system
  - Google Maps Scraping
    - Developed a Google Maps Scrapper to scrape the computer repair centers for recommendation
  - Reviews automation
    - Suggest the video link and a summary of a YouTube review regarding a recommended device to the user

ASR – CTC Approach(Based on Deep Speech 2)

- Connectionist temporal classification used where

It's a model with RNN and CNN and trained on LJ speech dataset

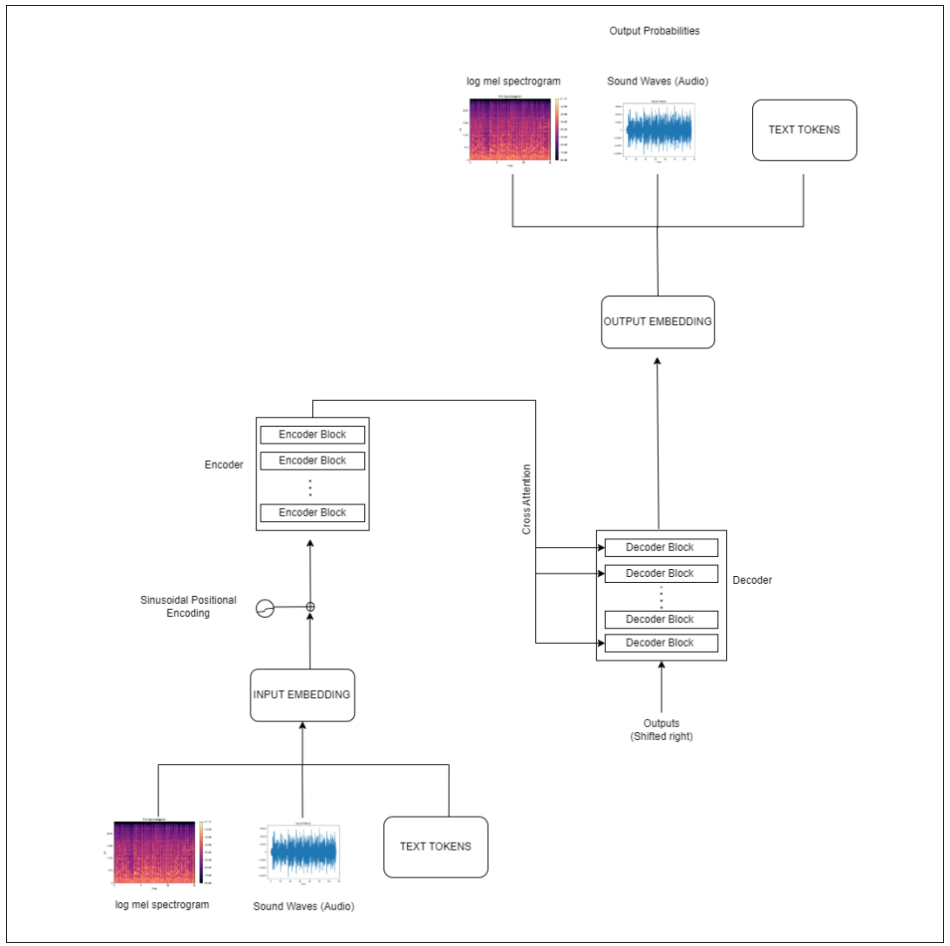
```python
def build_model(input_dim, output_dim, rnn_layers=5, rnn_units=128):

    # Model's input
    input_spectrogram = layers.Input((None, input_dim), name="input")
    # Expand the dimension to use 2D CNN.
    x = layers.Reshape((-1, input_dim, 1), name="expand_dim")(input_spectrogram)
    # Convolution layer 1
    x = layers.Conv2D(
        filters=32,
        kernel_size=[11, 41],
        strides=[2, 2],
        padding="same",
        use_bias=False,
        name="conv_1",
    )(x)
    x = layers.BatchNormalization(name="conv_1_bn")(x)
    x = layers.ReLU(name="conv_1_relu")(x)
    # Convolution layer 2
    x = layers.Conv2D(
        filters=32,
        kernel_size=[11, 21],
        strides=[1, 2],
        padding="same",
        use_bias=False,
        name="conv_2",
    )(x)
    x = layers.BatchNormalization(name="conv_2_bn")(x)
    x = layers.ReLU(name="conv_2_relu")(x)
    # Reshape the resulted volume to feed the RNNs layers
    x = layers.Reshape((-1, x.shape[-2] * x.shape[-1]))(x)
    # RNN layers
    for i in range(1, rnn_layers + 1):
        recurrent = layers.GRU(
            units=rnn_units,
            activation="tanh",
            recurrent_activation="sigmoid",
            use_bias=True,
            return_sequences=True,
            reset_after=True,
            name=f"gru_{i}",
        )
        x = layers.Bidirectional(
            recurrent, name=f"bidirectional_{i}", merge_mode="concat"
        )(x)
        if i < rnn_layers:
            x = layers.Dropout(rate=0.5)(x)
    # Dense layer
    x = layers.Dense(units=rnn_units * 2, name="dense_1")(x)
    x = layers.ReLU(name="dense_1_relu")(x)
    x = layers.Dropout(rate=0.5)(x)
    # Classification layer
    output = layers.Dense(units=output_dim + 1, activation="softmax")(x)
    # Model
    model = keras.Model(input_spectrogram, output, name="DeepSpeech_2")
    # Optimizer
    opt = keras.optimizers.Adam(learning_rate=1e-4)
    # Compile the model and return
    model.compile(optimizer=opt, loss=CTCLoss)
    return model


# Get the model
model = build_model(
    input_dim=fft_length // 2 + 1,
    output_dim=char_to_num.vocabulary_size(),
    rnn_units=512,
)
```

- Downsides of this model was it requires high computational power and less accurate even after running 20 epochs

```
1/1 [==============================] - 1s 685ms/step
1/1 [==============================] - 1s 728ms/step
1/1 [==============================] - 1s 614ms/step
1/1 [==============================] - 1s 750ms/step
1/1 [==============================] - 1s 680ms/step
1/1 [==============================] - 1s 675ms/step
1/1 [==============================] - 1s 700ms/step
1/1 [==============================] - 1s 739ms/step
1/1 [==============================] - 1s 607ms/step
1/1 [==============================] - 1s 621ms/step
--------------------------------------------------------------------------------
Word Error Rate: 0.2681
--------------------------------------------------------------------------------
Target    : and you have to establish adequate circulation
Prediction: and you havfet establish adequate cartulation
--------------------------------------------------------------------------------
Target    : and most of the records which have come from his time speak chiefly of his deeds of piety
Prediction: and most of the records which ove come from his time sp chiefly of his deds of peity
--------------------------------------------------------------------------------
369/369 [==============================] - 1322s 4s/step - loss: 30.1977 - val_loss: 44.8476
Epoch 20/20
1/1 [==============================] - 1s 719ms/step loss: 28.91
1/1 [==============================] - 1s 724ms/step
1/1 [==============================] - 1s 739ms/step
1/1 [==============================] - 1s 618ms/step
1/1 [==============================] - 1s 645ms/step
1/1 [==============================] - 1s 665ms/step
1/1 [==============================] - 1s 731ms/step
1/1 [==============================] - 1s 672ms/step
1/1 [==============================] - 1s 671ms/step
1/1 [==============================] - 1s 724ms/step
1/1 [==============================] - 1s 685ms/step
1/1 [==============================] - 1s 764ms/step
1/1 [==============================] - 1s 677ms/step
1/1 [==============================] - 1s 650ms/step
1/1 [==============================] - 1s 669ms/step
1/1 [==============================] - 1s 686ms/step
1/1 [==============================] - 1s 602ms/step
1/1 [==============================] - 1s 663ms/step
1/1 [==============================] - 1s 715ms/step
1/1 [==============================] - 1s 624ms/step
1/1 [==============================] - 1s 734ms/step
1/1 [==============================] - 1s 639ms/step
1/1 [==============================] - 1s 658ms/step
1/1 [==============================] - 1s 670ms/step
1/1 [==============================] - 1s 637ms/step
1/1 [==============================] - 1s 655ms/step
1/1 [==============================] - 1s 740ms/step
1/1 [==============================] - 1s 606ms/step
1/1 [==============================] - 1s 706ms/step
1/1 [==============================] - 1s 732ms/step
1/1 [==============================] - 1s 677ms/step
1/1 [==============================] - 1s 652ms/step
1/1 [==============================] - 1s 683ms/step
1/1 [==============================] - 1s 617ms/step
1/1 [==============================] - 1s 744ms/step
1/1 [==============================] - 1s 678ms/step
1/1 [==============================] - 1s 679ms/step
1/1 [==============================] - 1s 670ms/step
1/1 [==============================] - 1s 735ms/step
1/1 [==============================] - 1s 606ms/step
1/1 [==============================] - 1s 628ms/step
--------------------------------------------------------------------------------
Word Error Rate: 0.2609
--------------------------------------------------------------------------------
Target    : all the misdemeanants whatever their offense were lodged in this chapel ward
Prediction: all the misdemeanants whatever ther offense were lodge in this chapel ward
--------------------------------------------------------------------------------
Target    : to inquire into and report upon the several jails and houses of correction in the counties cities and corporate towns within england and wales
Prediction: to equire into an report upon the several jails and houses of cerection in the counties cities and corporate tounds withan england and wail
--------------------------------------------------------------------------------
369/369 [==============================] - 1376s 4s/step - loss: 28.9184 - val_loss: 45.6976
```

ASR – Seq2Seq Approach

- Used Whisper model which was released in end of September 2022 where it has been trained for 680,000 hours of audio data and fined tuned this whisper model for my requirements with “Commen Accent” dataset.



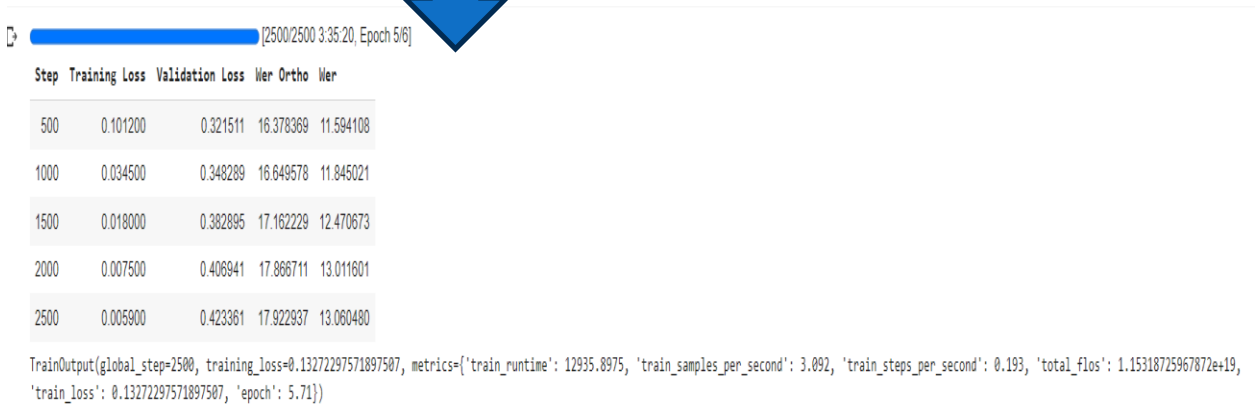
Best Approach

CTC Approach	Seq2Seq Approach
High usage of computational power	Comparatively low usage of computational power
High rate of spelling errors	Low rate of Spelling errors
Only use Encoder	Use both decoder and encoder (Can enhance the ASR for multiple languages)
Word Accuracy is Low	Word Accuracy is High
Difficult to handle High robustness, Noise inputs	High robustness, Noise inputs can be handled

Due to the above comparison, I had selected the seq2seq architecture model to fit for the chatbot

Model Results

CTC – Model based on Deep Speech2	Seq2Seq Whisper finetuned model
WER = 0.26 (26%) WAcc = 1- 0.26 = 0.74 (74 %)	WER = 0.13 (13%) Wacc = 1 – 0.13 = 0.87 (87 %)



Seq2Seq - Model Testing

- 2 models were built for the Seq2Seq approach.

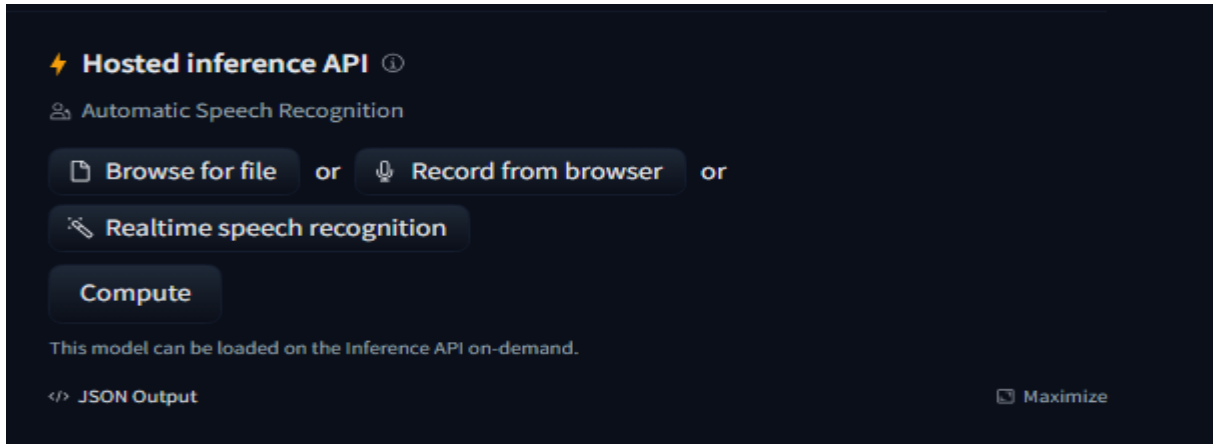
Model 1 Model with 500 steps up to 8 epochs	Model 2 Model with 2500 steps up to 6 epochs
Training loss - 0.0019 Epoch - 7.94 Step - 500 Validation Loss - 0.6059 WER Ortho - 22.9616 WER - 19.0970	Training loss - 0.0059 Epoch – 5.71 Step - 2500 Validation Loss – 0.4234 WER Ortho - 17.9229 WER - 13.0605

Seq2Seq - Model Testing

- Created custom Audio clips to test the model.

Audio	Transcript
	I need a laptop to do my personal things and it should have a RTX 3060 VGA and 12GB RAM so I can do my stuff without any lag

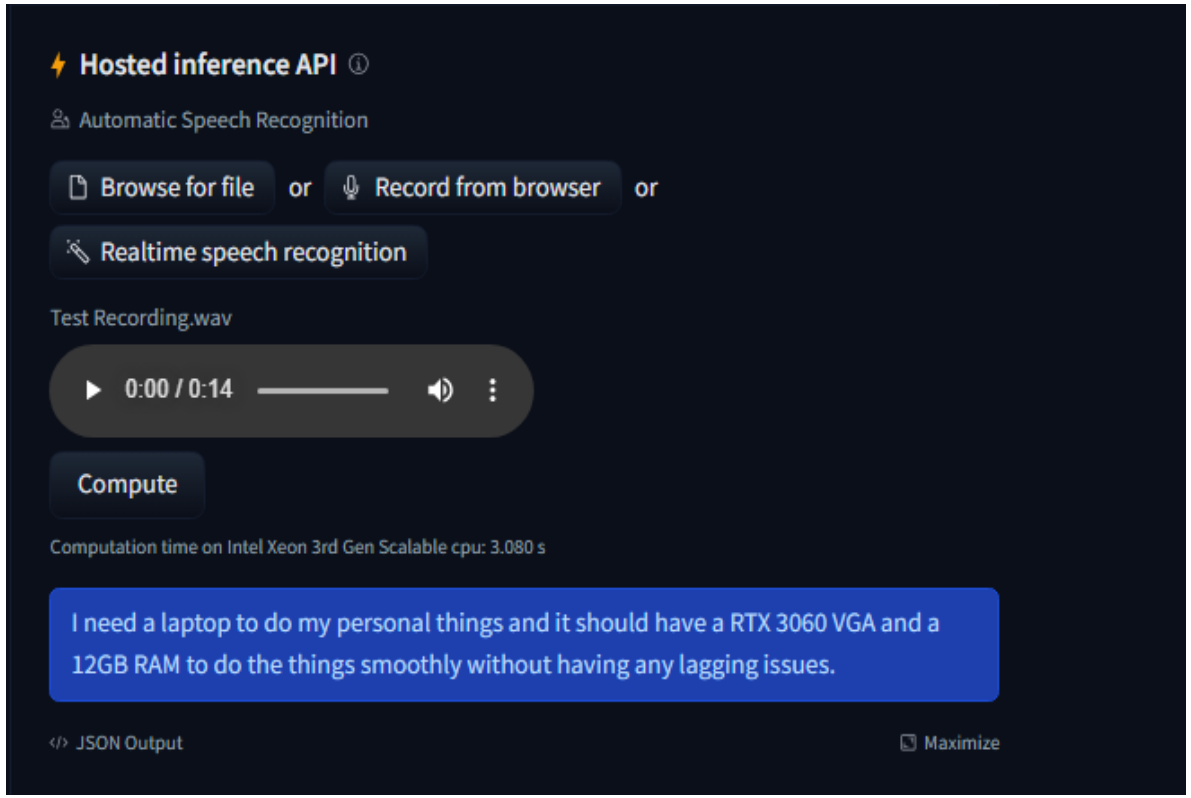
- Since both models were deployed in the Hugging Face API, tested those models by uploading the transcript for the API interface.



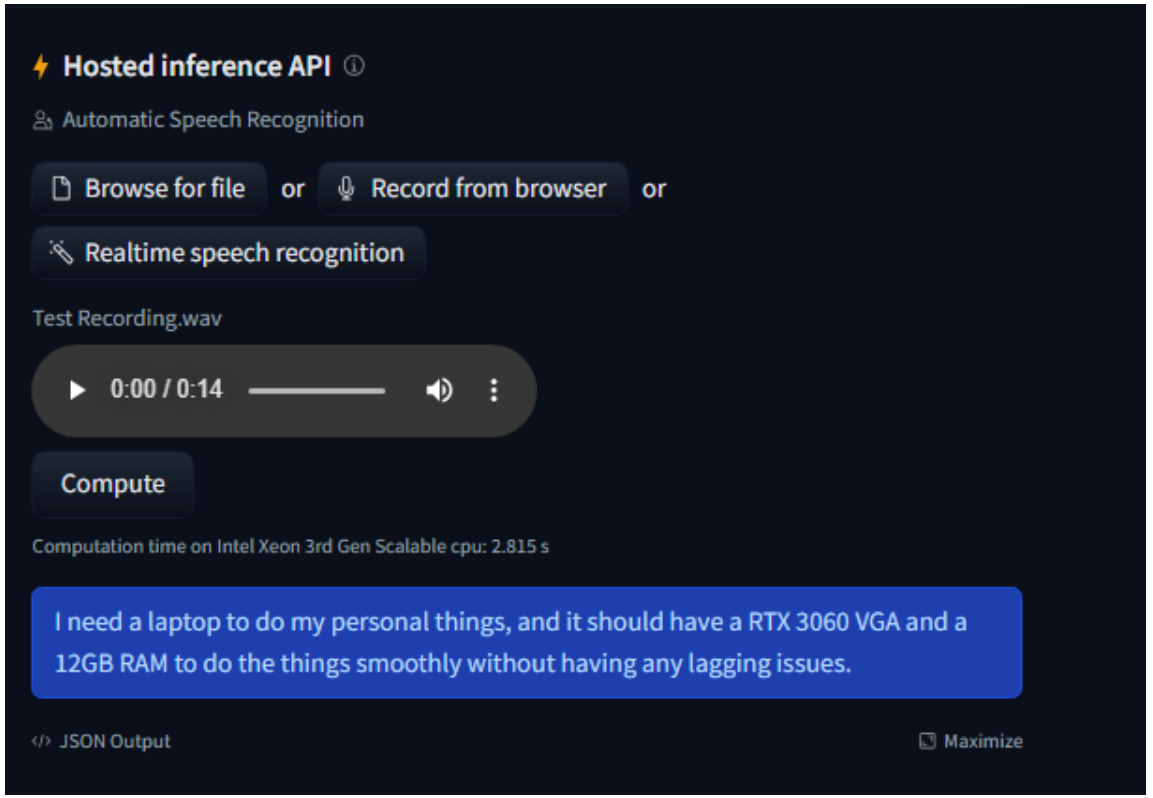
Seq2Seq Approach Model Results

When uploading an audio clip

- Model 1 Results



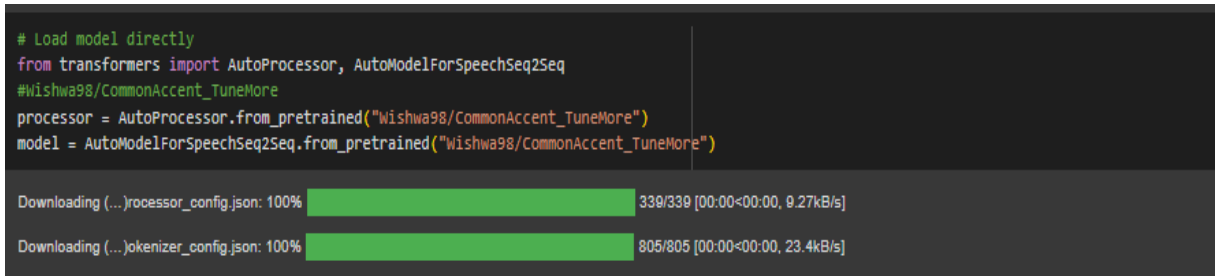
- Model 2 Results



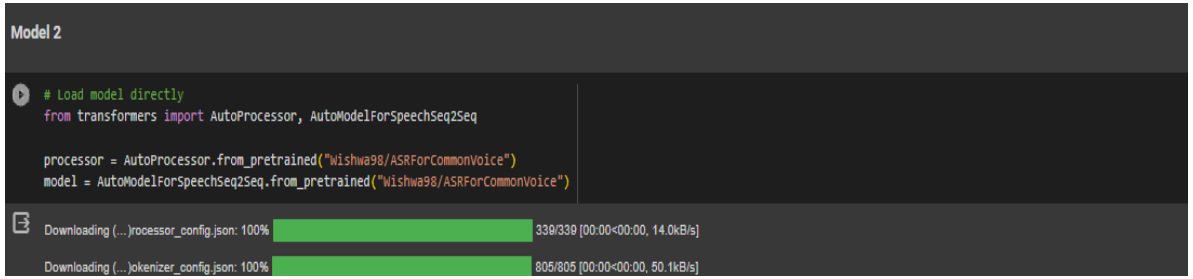
Seq2Seq Approach Model Results

For Real-time Voice recognition

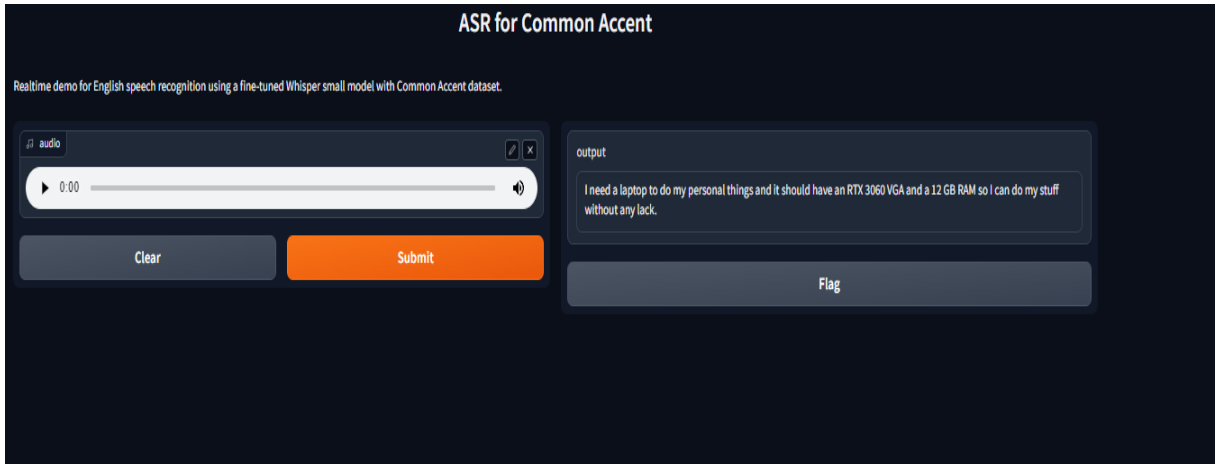
- Loading model 1



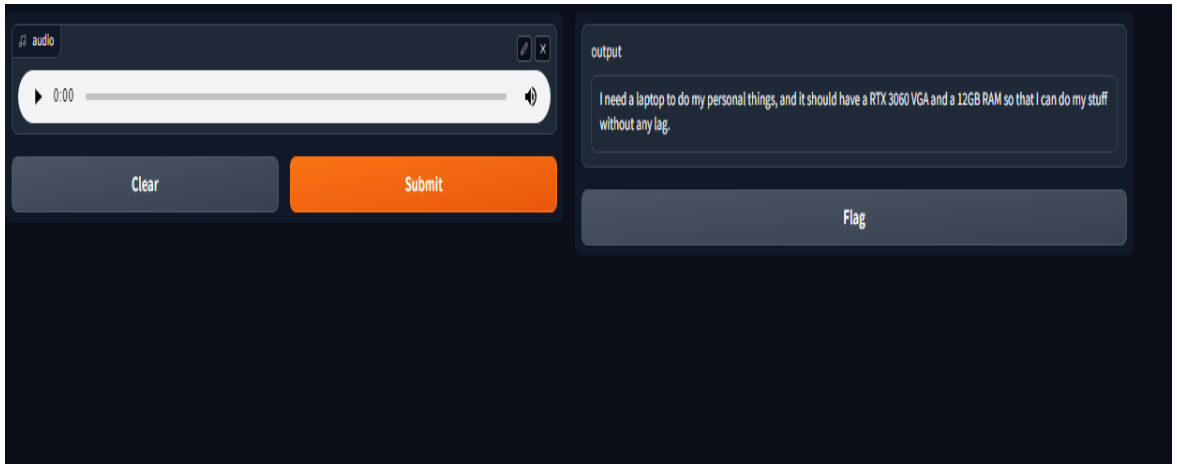
- Loading model 2



- Model 1 Results



- Model 2 Results



Evaluation Metrices

There are different word error categories as Substitution, Insertion Deletion in both word level and character level Here we used the Word level, which calculates the substitutions insertions, and deletions on a word level These errors are annotated on a word-by-word basis,

S=substitute

D=Delete

I=Insertion

N= Total number of words

$$WER = \frac{S + I + D}{N}$$

Word Error Rate is an error matrix that can be convert into accuracy matrix

$$W_{Acc} = 1 - WER$$

Seq2Seq Approach Model Results

- **Expected Text – " I need a laptop to do my personal things, and it should have a RTX 3060 VGA and a 12GB RAM so that I can do my stuff without any lag "**

Predicted Text (Model 1)	Predicted Text (Model 2)
"I need a laptop to do my personal things and it should have an RTX 3060 VGA and a 12GB RAM so I can do my stuff without any lack."	"I need a laptop to do my personal things, and it should have an RTX 3060 VGA and a 12GB RAM so that I can do my stuff without any lag."

- Can clearly see in the model 1 word "That" has been deleted and the word "lag" has been substituted with word "lack"

- Due to the model evaluation results and the Realtime testing results , Selected the model 2 to implement In the chatbot interface.