Dr.Mahalingam College of Engineering & Technology
Department of Artificial Intelligence & Data Science
19ADPN6401 – Mini Project
Final Review

Team Number: 23BADA016
Domain: Data Science

Title: **Multilingual Code Switching ASR(Automatic Speech Recognition)**

Guide :Ms D.C Kiruthikka,AP/AD
Ms R. Geetha Rajakumari, AP/AD

Team Members:	Subash B	(727622BAD037)
	Dharani S	(727622BAD071)
	NaveenRaj M	(727622BAD115)

Contents

Problem Statement:

- The problem is to develop an automatic speech recognition (ASR) system tailored for Indian languages, focusing on Tamil and English.
- The system will use a sequence-to-sequence (Seq2Seq) framework within end-to-end (E2E) models to accurately transcribe audio containing a mix of Tamil and English into English text.
- Key components include handling multilingual input, implementing Seq2Seq models, using E2E architecture for direct mapping, processing code-switching scenarios, and ensuring precise transcription output in English.

Literature Identified and Findings

Literatures Identified	Findings	Result
Exploration of End-to-End Framework for Code-Switching Speech Recognition Task: Challenges and Enhancements	Experiments on Hindi to English code-switching	Word Error Rate : 29.79%,
End-to-end multilingual automatic speech recognition for less-resourced languages: the case of four Ethiopian languages.	ASR system training with DNN(Deep Neural Network) is used.	WER : 29.83%
Speaker and Language Change Detection using Wav2vec2 and WhisperTijn Berns, Nik Vaessen, David A. van Leeuwen	ASR and SCD tasks in a single model. And found 10% of WER	WER= 11.7%

Literatures Identified	Findings	Result
Dual Script E2E framework for Multilingual and Code-Switching ASR Mari Ganesh Kumar, Jom Kuriakose, Anand Thyagachandran, Arun Kumar A, Ashish Seth, Lodagala Durga Prasad, Saish Jaiswal, Anusha Prakash, Hema Murthy	Proposed an ASR of phoneme-level common label set (CLS) for native language characters during training.	Word Error Rate approximately 5% to 6% improvements.
Crossing language identification: Multilingual ASR framework based on semantic dataset creation & Wav2Vec 2.0	Trained the Wav2vec 2.0 XLSR53 model on the datasets and assess its performance utilizing Character Error Rate (CER) and Word Error Rate(WER) metrics.	Language- Russian& Portuguese WER:26.22%

Performance Evaluation Metrics:

The evaluation of automatic speech recognition (ASR) systems has predominantly centered around the utilization of error-based metrics, such as the Word Error Rate (WER) And Character Error Rate (CER), Accuracy,Precision,F1 Score.

Objective:

- The objective is to develop an automatic speech recognition (ASR) system specifically designed for Indian languages, with a primary focus on Tamil and English.

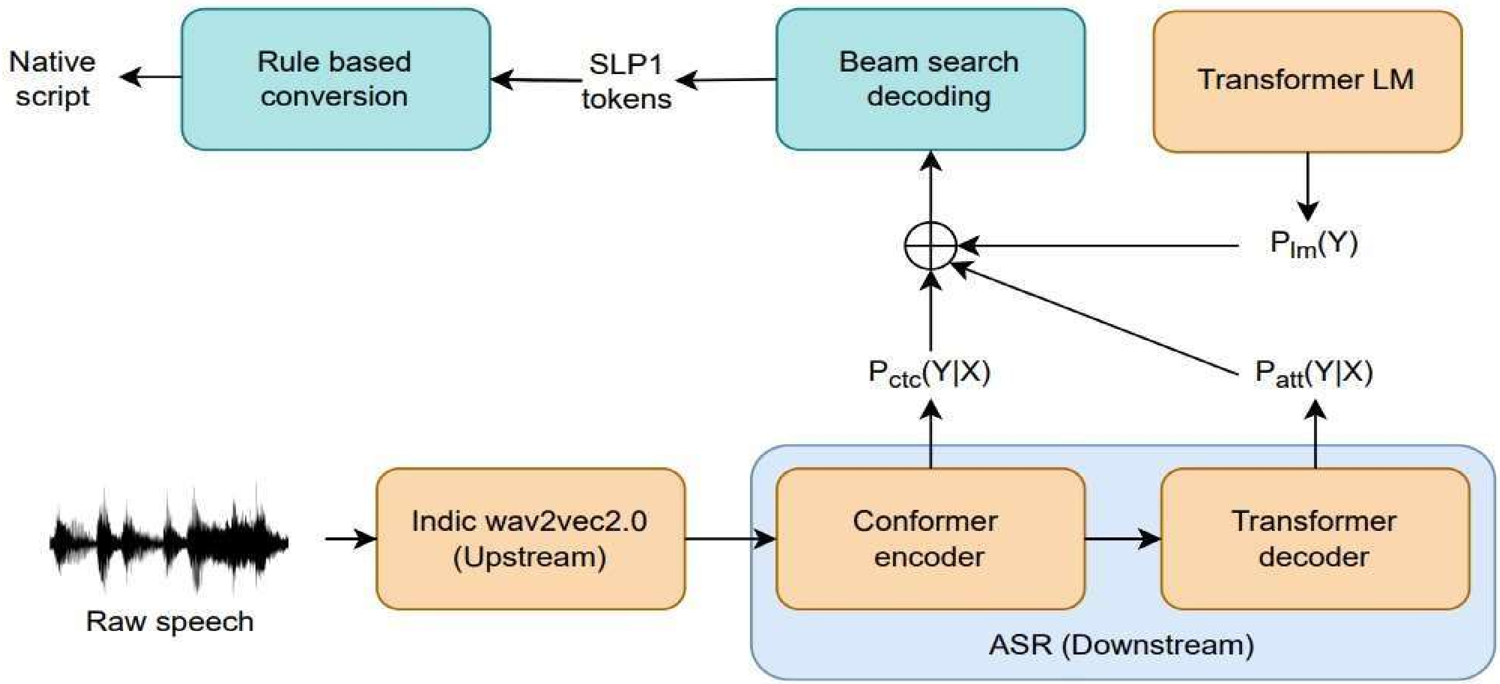
- This system will utilize a sequence-to-sequence (Seq2Seq) framework within end-to-end (E2E) models to accurately transcribe audio that contains a mixture of Tamil and English into English text.

Objective

- Key components of this objective involve managing multilingual input, deploying Seq2Seq models, leveraging E2E architecture for direct mapping, handling code-switching situations, and ensuring precise transcription output in English.

- The research aims to enhance ASR systems to effectively handle code-switching between these languages, improving overall accuracy and efficiency in recognizing mixed-language speech inputs.

Existing Block Diagram:



Drawbacks Of Existing System

- Minimum WER, CER, Accuracy, F1 Score
- Code-Switching - multiple languages are within a sentence
- Unique Scripts
- Noise and Environmental Conditions
- Code-switching Pattern- adapting to variations is highly challenging
- Multilingual Diversity

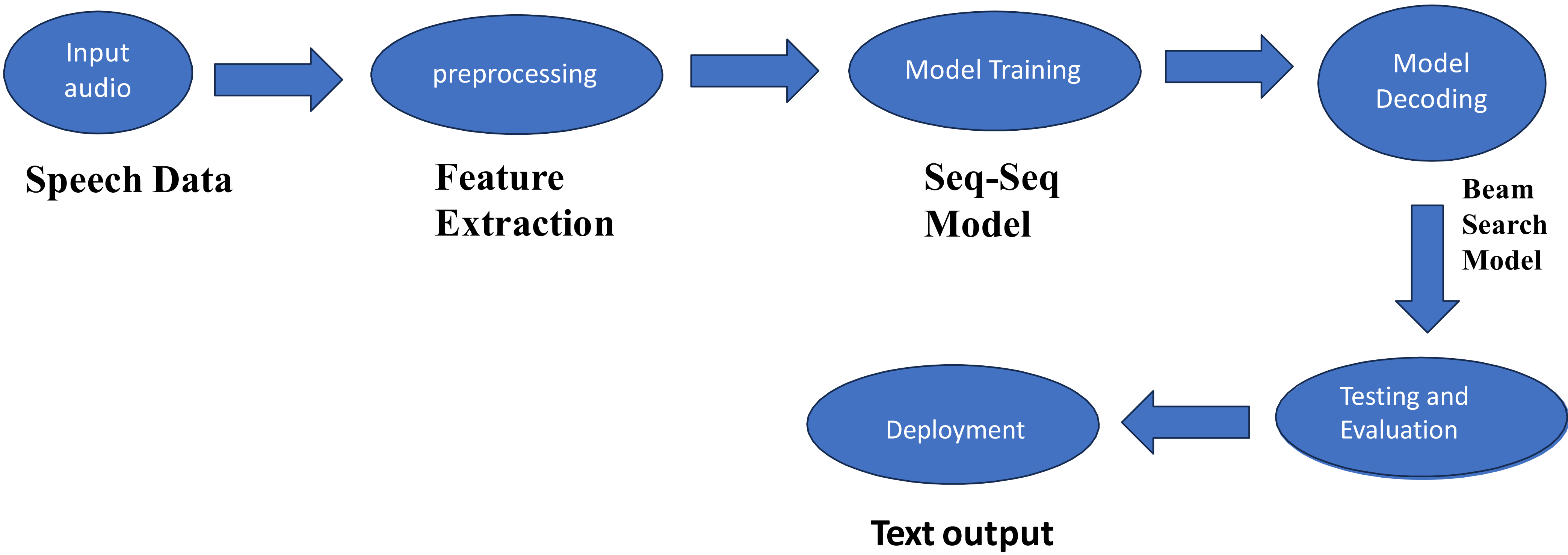
Proposed system:

- In existing systems, the availability of datasets especially for European languages like Russian, Portuguese, and Spanish.
- This proposed system includes Indian languages Tamil, Sanskrit, English, and other languages.
- This application can transcript information into English text when Tamil-English audio is given as input.

Proposed system

- Advanced Neural Network Architecture(RNN+ LSTM).
- Multilingual capabilities for handling multiple linguistic environments.
- Robustness in Speech Quality.
- Adaptation and personalization to user voice, accent, and style of speaking over time.
- User Experience which Focuses on creating a seamless and intuitive user interface.
- Handling complex commands and providing useful feedback.

Proposed Block Diagram:



Module Description

- ❑ **Audio Input (Speech Data):** Raw speech data as input.

- ❑ **Preprocessing**
- **Noise Reduction:** Remove background noise and enhance audio quality.
- **Feature Extraction:** Converts the audio input into a feature representation suitable for the model.(spectrogram segments,pitch, amplitude,sounds)

- ❑ **End-to-End (E2E) Framework Setup:**Implementing a Seq2Seq model using recurrent neural networks (RNNs), such as LSTM or GRU cells, for direct mapping of audio features to text sequences. The model architecture includes encoder and decoder components for input processing and output generation.

Module Description

- ❑ **Attention Mechanisms:** Integrating attention mechanisms within the Seq2Seq model to enhance context understanding.
- Improve alignment between audio inputs and transcriptions.

- ❑ **Decoding (Beam Search Model):** Searches for the most likely sequence of words given the model's output.

- ❑ **Post-Processing:** Applying post-processing techniques like language model integration, spell checking, and punctuation normalization to refine the final transcriptions and improve readability.

Module Description

❑ **Evaluation and Metrics:**

- Performance Metrics: WER,CER and accuracy to assess transcription quality.
- Cross-Validation

❑ **Integration and Deployment:**Integrating the trained ASR model into applications such as voice assistants, transcription services.

Requirements

- Datasets of Indian languages
- Python 3 or above.
- Sypder 5.0 version or above.
- Anaconda python.
- Laptop or pc over intel i5 11th gen processor for fast processing.

Results And Discussion

min.ipynb - Colab

colab.research.google.com/drive/1XJ1hxCuhaKeeV8y8bZ0ku4r5NI5EUqHl#scrollTo=luIQEAlltC-7

min.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text Connect T4 Colab AI

```
print(tokenizer_cs.word_index)
```

```
{'<unk>': 1, 'indru': 2, 'netru': 3, 'kadigaram': 4, 'naalai': 5, 'illa': 6, 'parunga': 7, 'chinna': 8, 'periya': 9, 'alagu': 10, 'nalla': 11, 'ketta': 12, 'nandri': 13, 'yedho': 14,
```

```
print("Decoder vocabulary:", tokenizer_en.word_index)
```

```
Decoder vocabulary: {'<unk>': 1, 'today': 2, 'yesterday': 3, 'clock': 4, 'naalai': 5, 'no/not': 6, 'see': 7, 'small': 8, 'big': 9, 'beautiful': 10, 'good': 11, 'bad': 12, 'thankyou':
```

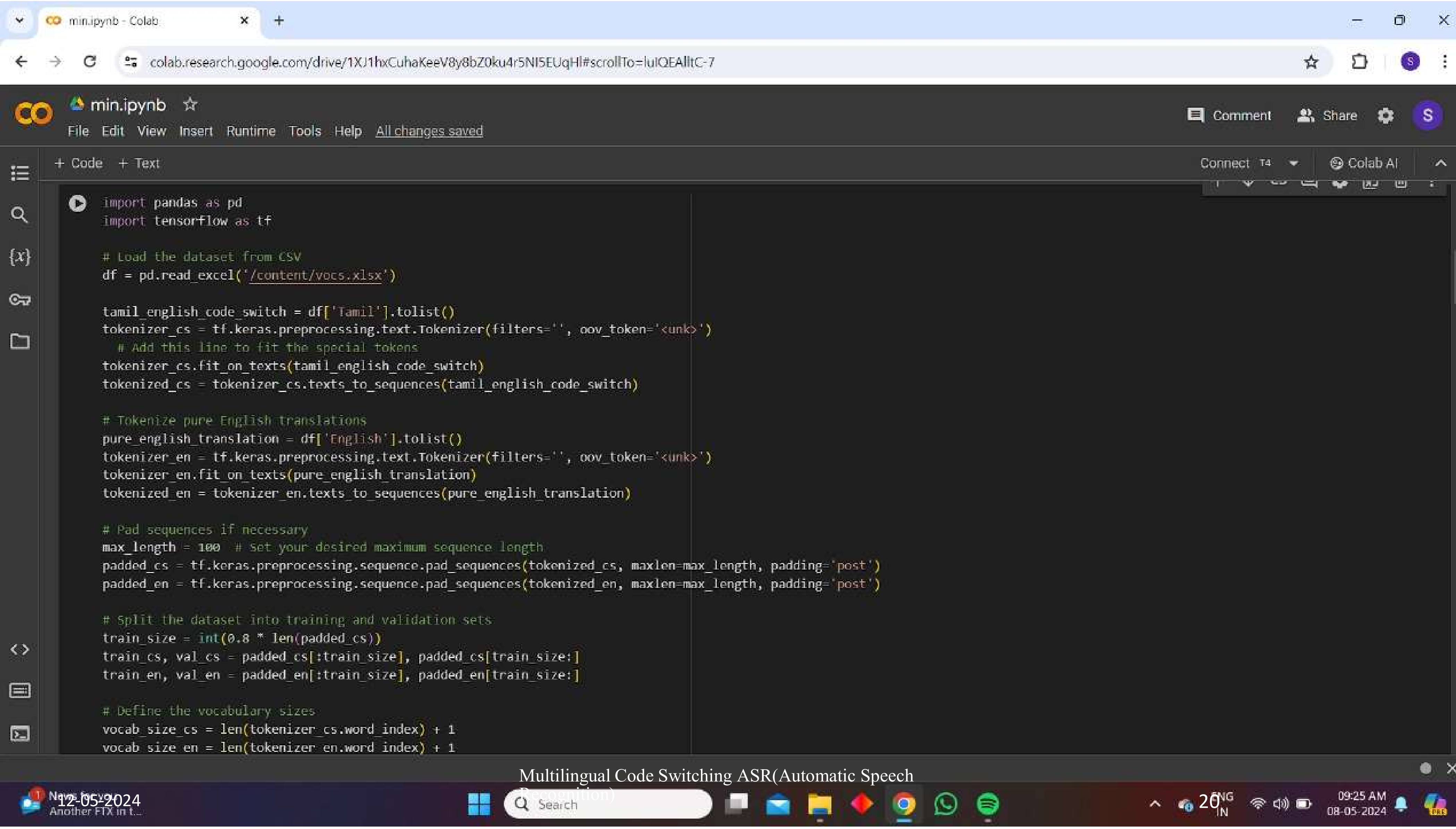
```
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Input, Embedding, LSTM, Dense, Attention, Concatenate

# Define embedding dimensions and encoder units
embedding_dim = 256
encoder_units = 512
decoder_units = 512

# Define the encoder
# Define the encoder input layer with the correct shape
encoder_input = Input(shape=(padded_cs.shape[1],))
encoder_embedding = Embedding(vocab_size_cs, embedding_dim, input_length=max_length)(encoder_input)
encoder_lstm = LSTM(encoder_units, return_sequences=True, return_state=True)
encoder_outputs, state_h, state_c = encoder_lstm(encoder_embedding)
encoder_states = [state_h, state_c]

# Define the decoder
decoder_input = Input(shape=(None,))
decoder_embedding = Embedding(vocab_size_en, embedding_dim)
```

News for you
Another FTX in t...

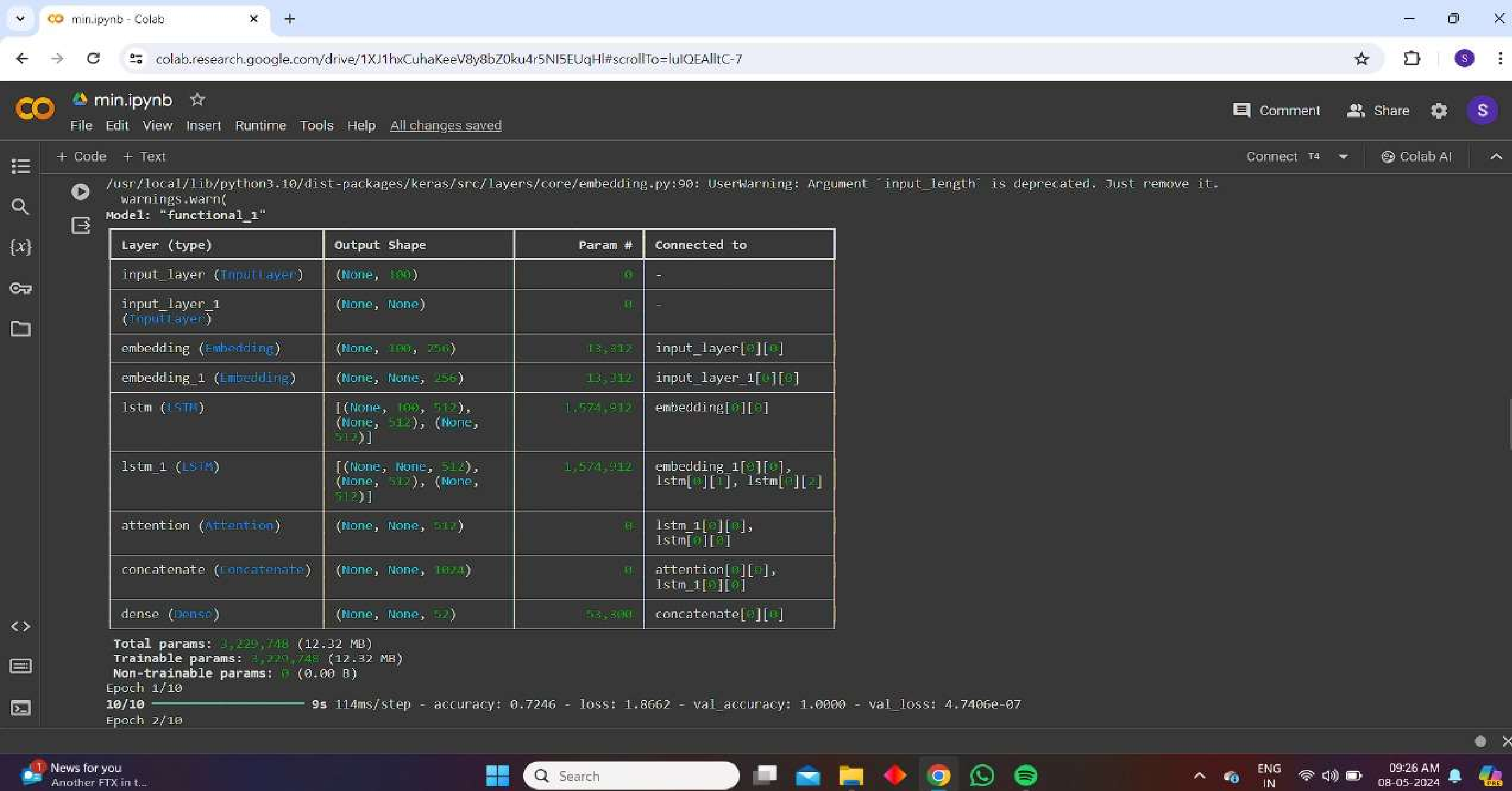
Search

ENG IN 09:25 AM 08-05-2024

Results And Discussion



Results And Discussion




```
/usr/local/lib/python3.10/dist-packages/keras/src/layers/core/embedding.py:90: UserWarning: Argument `input_length` is deprecated. Just remove it.
  warnings.warn(
Model: "functional_1"
```

Layer (type)	Output Shape	Param #	Connected to
input_layer (InputLayer)	(None, 100)	0	-
input_layer_1 (InputLayer)	(None, None)	0	-
embedding (Embedding)	(None, 100, 256)	13,312	input_layer[0][0]
embedding_1 (Embedding)	(None, None, 256)	13,312	input_layer_1[0][0]
lstm (LSTM)	[(None, 100, 512), (None, 512), (None, 512)]	1,574,912	embedding[0][0]
lstm_1 (LSTM)	[(None, None, 512), (None, 512), (None, 512)]	1,574,912	embedding_1[0][0], lstm[0][1], lstm[0][2]
attention (Attention)	(None, None, 512)	0	lstm_1[0][0], lstm[0][0]
concatenate (Concatenate)	(None, None, 1024)	0	attention[0][0], lstm_1[0][0]
dense (Dense)	(None, None, 52)	53,300	concatenate[0][0]

```
 Total params: 3,229,748 (12.32 MB)
 Trainable params: 3,229,748 (12.32 MB)
 Non-trainable params: 0 (0.00 B)
Epoch 1/10
10/10 ━━━━━━━━━━━━━━━━━━━━ 9s 114ms/step - accuracy: 0.7246 - loss: 1.8662 - val_accuracy: 1.0000 - val_loss: 4.7406e-07
Epoch 2/10
```

min.ipynb - Colab

colab.research.google.com/drive/1XJ1hxCuhaKeeV8y8bZ0ku4r5NI5EUqHl#scrollTo=luIQEAlltC-7

min.ipynb

File Edit View Insert Runtime Tools Help All changes saved

```
encoder_model.summary()
```

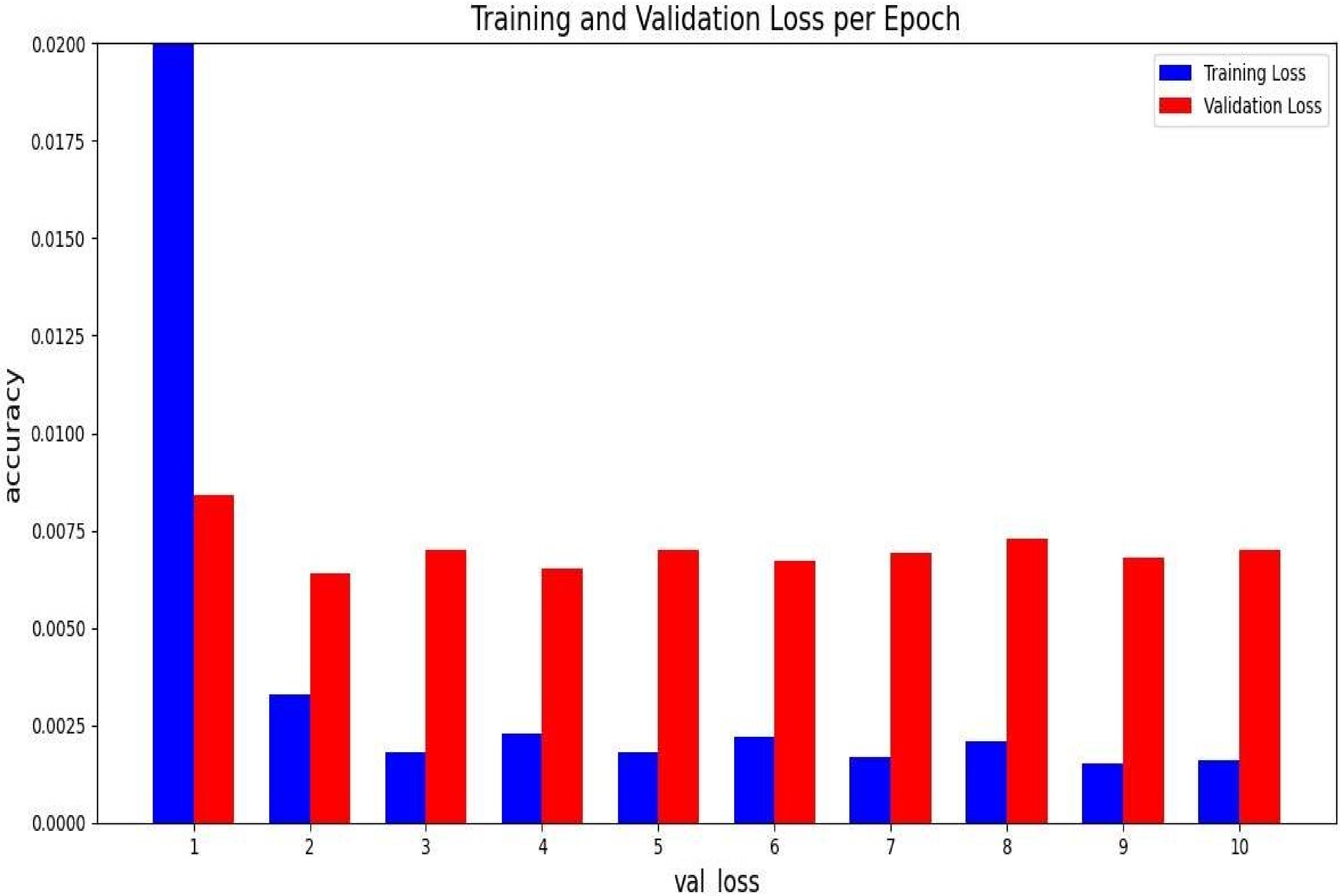
Model: "functional_3"

Layer (type)	Output Shape	Param #
input_layer (InputLayer)	(None, 100)	0
embedding (Embedding)	(None, 100, 256)	13,312
lstm (LSTM)	[(None, 100, 512), (None, 512), (None, 512)]	1,574,912

```
Total params: 1,588,224 (6.06 MB)
Trainable params: 1,588,224 (6.06 MB)
Non-trainable params: 0 (0.00 B)
```

```
# Tokenize the input sentence
input_sentence = "Sometimes makkal drinks tea"
tokenized_sequence = tokenizer_cs.texts_to_sequences([input_sentence])[0]

# Translate token IDs into words using the vocabulary
words = [tokenizer_en.index_word[token_id] for token_id in tokenized_sequence]


print(tokenized_sequence)

# Print the translated words
print("Translated Sentence:", ' '.join(words))
```

```
[1, 26, 1, 16]
Translated Sentence: <unk> people <unk> tea
```



Training and Validation Loss per Epoch
Training Loss
Validation Loss
accuracy
0.0200
0.0175
0.0150
0.0125
0.0100
0.0075
0.0050
0.0025
0.0000
1
2
3
4
5
6
7
8
9
10
val_loss

Online Course Details

NAME (ROLL NO)	COURSE NAME	PLATFORM	NO OF WEEKS/ HOUR	REMARKS/ STATUS
Subash B (727622BAD037)	SkillForge - Data Science	Courses	4 weeks/ 20 hours	Completed
Dharani S (727622BAD071)	SkillForge - Data Science	Courses	4 weeks/ 20 hours	Completed
Naveen Raj M (727622BAD115)	IBM - Python for Data Science	Courses	4 weeks/ 20 hours	Completed

Conclusion:

- The project's use of an End-to-End (E2E) Seq2Seq model for multilingual Automatic Speech Recognition (ASR) in Indian languages shows promising results, especially in accurately transcribing English and Tamil speech. Challenges like code-switching among Indian languages remain, highlighting the need for ongoing research.

- Overall, the project contributes valuable insights and sets a foundation for further innovation in multilingual ASR technologies.

References:

BASEPAPER :

- Crossing language identification: Multilingual ASR framework based on semantic dataset creation & Wav2Vec 2.0
- The data is publicly available at:

https://commonvoice.mozilla.org/en/datasets.

Crossing language identification: Multilingual ASR framework based on semantic dataset creation & Wav2Vec 2.0 - ScienceDirect

References:

<u>BOOK REFERENCE</u>

- **Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython**
  Book by *Wes McKinney*

References:

- Recent Advances in End-to-End Automatic Speech Recognition - arXiv.org. https://arxiv.org/pdf/2111.01690.

- ACOUSTIC MODEL FUSION FOR END-TO-END SPEECH RECOGNITION - arXiv.org. https://arxiv.org/pdf/2310.07062.pdf.

- E2EXf - How to Use E2E Transformers in Autosar - AutosarToday. https://www.autosartoday.com/posts/e2exf_-_how_to_use_e2e_transformers_in_autosar.

References:

- Recent Advances in End-to-End Automatic Speech Recognition. https://arxiv.org/abs/2111.01690.
- Dual Script E2E Framework for Multilingual and Code-Switching ASR. https://arxiv.org/pdf/2106.01400.
- Dual Script E2E framework for Multilingual and Code-Switching ASR: This paper discusses a framework for training ASR systems for Indian languages, addressing the challenges of code-mixing and script diversity[1].

1.