```
# Part 2: Implementation (Coding)
# 3. Model Setup
# a. Dataset Preprocessing
# Use a few English-French pairs:
data = [("hello", "bonjour"), ("how are you", "comment ça va"),
        ("thank you", "merci"), ("good morning", "bonjour"), ("i love you", "je t'aime")]
# b. Tokenization and Padding
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences

eng_sentences, fr_sentences = zip(*data)
eng_tokenizer = Tokenizer()
fr_tokenizer = Tokenizer()

eng_tokenizer.fit_on_texts(eng_sentences)
fr_tokenizer.fit_on_texts(fr_sentences)

eng_seq = eng_tokenizer.texts_to_sequences(eng_sentences)
fr_seq = fr_tokenizer.texts_to_sequences(fr_sentences)

max_eng_len = max(len(seq) for seq in eng_seq)
max_fr_len = max(len(seq) for seq in fr_seq)

eng_seq = pad_sequences(eng_seq, maxlen=max_eng_len, padding='post')
fr_seq = pad_sequences(fr_seq, maxlen=max_fr_len, padding='post')
# c. Encoder-Decoder with Attention (Simplified)
import tensorflow as tf
from tensorflow.keras.layers import Embedding, LSTM, Dense, Input, Attention
from tensorflow.keras.models import Model

# Parameters
vocab_eng = len(eng_tokenizer.word_index) + 1
vocab_fr = len(fr_tokenizer.word_index) + 1
embed_dim = 64
lstm_units = 64

# Encoder
encoder_inputs = Input(shape=(max_eng_len,))
enc_emb = Embedding(vocab_eng, embed_dim)(encoder_inputs)
encoder_lstm, state_h, state_c = LSTM(lstm_units, return_sequences=True, return_state=True)(enc_emb)

# Decoder
decoder_inputs = Input(shape=(max_fr_len,))
dec_emb = Embedding(vocab_fr, embed_dim)(decoder_inputs)
decoder_lstm_out, _, _ = LSTM(lstm_units, return_sequences=True, return_state=True)(dec_emb, initial_state=[state_h, state_c])

# Attention
attention = Attention()([decoder_lstm_out, encoder_lstm])
from tensorflow.keras.layers import Concatenate

concat = Concatenate(axis=-1)([decoder_lstm_out, attention])


# Output
outputs = Dense(vocab_fr, activation='softmax')(concat)

model = Model([encoder_inputs, decoder_inputs], outputs)
model.compile(optimizer='adam', loss='sparse_categorical_crossentropy')
model.summary()
# d. Training
import numpy as np
decoder_output = np.expand_dims(fr_seq, -1)
model.fit([eng_seq, fr_seq], decoder_output, epochs=100)
```

input_layer_2 (InputLayer)	(None, 3)	0	-
input_layer_3 (InputLayer)	(None, 3)	0	-
embedding_2 (Embedding)	(None, 3, 64)	640	input_layer_2[0]…
embedding_3 (Embedding)	(None, 3, 64)	512	input_layer_3[0]…
lstm_2 (LSTM)	[(None, 3, 64), (None, 64), (None, 64)]	33,024	embedding_2[0][0]
lstm_3 (LSTM)	[(None, 3, 64), (None, 64), (None, 64)]	33,024	embedding_3[0][0… lstm_2[0][1], lstm_2[0][2]
attention_1 (Attention)	(None, 3, 64)	0	lstm_3[0][0], lstm_2[0][0]
concatenate (Concatenate)	(None, 3, 128)	0	lstm_3[0][0], attention_1[0][0]
dense (Dense)	(None, 3, 8)	1,032	concatenate[0][0]

Total params: 68,232 (266.53 KB)
Trainable params: 68,232 (266.53 KB)
Non-trainable params: 0 (0.00 B)

```
Epoch 1/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 5s 5s/step - loss: 2.0846
Epoch 2/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 184ms/step - loss: 2.0741
Epoch 3/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 140ms/step - loss: 2.0636
Epoch 4/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 142ms/step - loss: 2.0530
Epoch 5/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 145ms/step - loss: 2.0420
Epoch 6/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 135ms/step - loss: 2.0306
Epoch 7/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 114ms/step - loss: 2.0187
Epoch 8/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 73ms/step - loss: 2.0060
Epoch 9/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 157ms/step - loss: 1.9924
Epoch 10/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 75ms/step - loss: 1.9779
Epoch 11/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 80ms/step - loss: 1.9621
Epoch 12/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 146ms/step - loss: 1.9450
Epoch 13/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 89ms/step - loss: 1.9263
Epoch 14/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 129ms/step - loss: 1.9059
Epoch 15/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 91ms/step - loss: 1.8835
Epoch 16/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 61ms/step - loss: 1.8588
Epoch 17/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 65ms/step - loss: 1.8316
Epoch 18/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 52ms/step - loss: 1.8016
Epoch 19/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 60ms/step - loss: 1.7686
Epoch 20/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 63ms/step - loss: 1.7323
Epoch 21/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 62ms/step - loss: 1.6926
Epoch 22/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 59ms/step - loss: 1.6493
Epoch 23/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 55ms/step - loss: 1.6025
Epoch 24/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 55ms/step - loss: 1.5527
Epoch 25/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 52ms/step - loss: 1.5005
```

```
Epoch 26/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 61ms/step - loss: 1.4471
Epoch 27/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 64ms/step - loss: 1.3943
Epoch 28/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 54ms/step - loss: 1.3441
Epoch 29/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 53ms/step - loss: 1.2989
Epoch 30/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 54ms/step - loss: 1.2607
Epoch 31/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 56ms/step - loss: 1.2307
Epoch 32/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 54ms/step - loss: 1.2090
Epoch 33/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 53ms/step - loss: 1.1943
Epoch 34/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 70ms/step - loss: 1.1840
Epoch 35/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 57ms/step - loss: 1.1753
Epoch 36/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 55ms/step - loss: 1.1656
Epoch 37/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 53ms/step - loss: 1.1533
Epoch 38/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 63ms/step - loss: 1.1376
Epoch 39/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 59ms/step - loss: 1.1186
Epoch 40/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 56ms/step - loss: 1.0970
Epoch 41/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 54ms/step - loss: 1.0737
Epoch 42/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 53ms/step - loss: 1.0496
Epoch 43/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 60ms/step - loss: 1.0254
Epoch 44/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 56ms/step - loss: 1.0019
Epoch 45/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 146ms/step - loss: 0.9794
Epoch 46/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 55ms/step - loss: 0.9581
Epoch 47/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 54ms/step - loss: 0.9383
Epoch 48/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 62ms/step - loss: 0.9199
Epoch 49/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 60ms/step - loss: 0.9027
Epoch 50/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 73ms/step - loss: 0.8867
Epoch 51/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 53ms/step - loss: 0.8717
Epoch 52/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 57ms/step - loss: 0.8572
Epoch 53/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 58ms/step - loss: 0.8430
Epoch 54/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 59ms/step - loss: 0.8286
Epoch 55/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 57ms/step - loss: 0.8137
Epoch 56/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 62ms/step - loss: 0.7981
Epoch 57/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 60ms/step - loss: 0.7818
Epoch 58/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 62ms/step - loss: 0.7651
Epoch 59/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 59ms/step - loss: 0.7482
Epoch 60/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 54ms/step - loss: 0.7317
Epoch 61/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 55ms/step - loss: 0.7157
Epoch 62/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 55ms/step - loss: 0.7005
Epoch 63/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 60ms/step - loss: 0.6862
Epoch 64/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 55ms/step - loss: 0.6727
Epoch 65/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 55ms/step - loss: 0.6597
Epoch 66/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 59ms/step - loss: 0.6472
Epoch 67/100
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 69ms/step - loss: 0.6349
Epoch 68/100
```