Ex. No. 9	Transformer model for a text classification task
Date of Exercise	**10/11/2025**

Aim:

To build a neural machine translation model using the Keras Functional API and evaluate translation quality
using BLEU scores.

Description:

Neural Machine Translation (NMT) is based on deep learning techniques that translate text from one language to another. BLEU scores measure the accuracy of the generated translations.

Code:

```
import tensorflow as tf
from tensorflow.keras.layers import Input, LSTM, Dense
from tensorflow.keras.models import Model
from nltk.translate.bleu_score import sentence_bleu
# Define model architecture
encoder_inputs = Input(shape=(None, 100))
encoder_lstm = LSTM(256, return_state=True)
encoder_outputs, state_h, state_c = encoder_lstm(encoder_inputs)
encoder_states = [state_h, state_c]
decoder_inputs = Input(shape=(None, 100))
decoder_lstm = LSTM(256, return_sequences=True, return_state=True)
```

```
decoder_outputs, _, _ = decoder_lstm(decoder_inputs, initial_state=encoder_states)
decoder_dense = Dense(100, activation='softmax')
decoder_outputs = decoder_dense(decoder_outputs)
# Define the model
nmt_model = Model([encoder_inputs, decoder_inputs], decoder_outputs)
nmt_model.compile(optimizer='adam', loss='categorical_crossentropy')
print(nmt_model.summary())
# BLEU score evaluation
def evaluate_bleu(reference, candidate):
return sentence_bleu([reference.split()], candidate.split())
reference_text = "hello world"
candidate_text = "hello earth"
print("BLEU Score:", evaluate_bleu(reference_text, candidate_text))
```

Sample Output:

```
Model: "functional"
```

Layer (type)	Output Shape	Param #	Connected to
input_layer (InputLayer)	(None, None, 100)	0	-
input_layer_1 (InputLayer)	(None, None, 100)	0	-
lstm (LSTM)	[(None, 256), (None, 256), (None, 256)]	365,568	input_layer[0][0]
lstm_1 (LSTM)	[(None, None, 256), (None, 256), (None, 256)]	365,568	input_layer_1[0]… lstm[0][1], lstm[0][2]
dense (Dense)	(None, None, 100)	25,700	lstm_1[0][0]

```
Total params: 756,836 (2.89 MB)

Trainable params: 756,836 (2.89 MB)

Non-trainable params: 0 (0.00 B)

None
```

```
candidate_text = "hello earth"
print("BLEU Score:", evaluate_bleu(reference_text, candidate_text))
```

✓ 0.0s

```
BLEU Score: 1.5319719891192393e-231
/home/franz/Documents/Lab/.venv/lib/python3.12/site-packages/nltk/translate/bleu_score.py:577: UserWarning:
The hypothesis contains 0 counts of 2-gram overlaps.
Therefore the BLEU score evaluates to 0, independently of
how many N-gram overlaps of lower order it contains.
Consider using lower n-gram order or use SmoothingFunction()
  warnings.warn(_msg)
/home/franz/Documents/Lab/.venv/lib/python3.12/site-packages/nltk/translate/bleu_score.py:577: UserWarning:
The hypothesis contains 0 counts of 3-gram overlaps.
Therefore the BLEU score evaluates to 0, independently of
how many N-gram overlaps of lower order it contains.
Consider using lower n-gram order or use SmoothingFunction()
  warnings.warn(_msg)
/home/franz/Documents/Lab/.venv/lib/python3.12/site-packages/nltk/translate/bleu_score.py:577: UserWarning:
The hypothesis contains 0 counts of 4-gram overlaps.
Therefore the BLEU score evaluates to 0, independently of
how many N-gram overlaps of lower order it contains.
Consider using lower n-gram order or use SmoothingFunction()
  warnings.warn(_msg)
```

Result:

The above experiment of Transformer model for a text classification task is done successfully and the output is been obtained