In [1]:
```
pip install tensorflow numpy pandas scikit-learn
```

```
Requirement already satisfied: tensorflow in c:\users\dell\anaconda3\lib\site-packages (2.18.0)
Requirement already satisfied: numpy in c:\users\dell\anaconda3\lib\site-packages (1.26.4)
Requirement already satisfied: pandas in c:\users\dell\anaconda3\lib\site-packages (1.5.3)
Requirement already satisfied: scikit-learn in c:\users\dell\anaconda3\lib\site-packages (1.2.1)
Requirement already satisfied: tensorflow-intel==2.18.0 in c:\users\dell\anaconda3\lib\site-packages (from tensorflow)
(2.18.0)
Requirement already satisfied: typing-extensions>=3.6.6 in c:\users\dell\anaconda3\lib\site-packages (from tensorflow-
intel==2.18.0->tensorflow) (4.12.2)
Requirement already satisfied: wrapt>=1.11.0 in c:\users\dell\anaconda3\lib\site-packages (from tensorflow-intel==2.1
8.0->tensorflow) (1.14.1)
Requirement already satisfied: opt-einsum>=2.3.2 in c:\users\dell\anaconda3\lib\site-packages (from tensorflow-intel==
2.18.0->tensorflow) (3.4.0)
Requirement already satisfied: setuptools in c:\users\dell\anaconda3\lib\site-packages (from tensorflow-intel==2.18.0-
>tensorflow) (65.6.3)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in c:\users\dell\anaconda3\lib\site-packages (from tensorflow-intel
==2.18.0->tensorflow) (1.68.1)
Requirement already satisfied: tensorboard<2.19,>=2.18 in c:\users\dell\anaconda3\lib\site-packages (from tensorflow-i
ntel==2.18.0->tensorflow) (2.18.0)
Requirement already satisfied: requests<3,>=2.21.0 in c:\users\dell\anaconda3\lib\site-packages (from tensorflow-intel
==2.18.0->tensorflow) (2.28.1)
Requirement already satisfied: h5py>=3.11.0 in c:\users\dell\anaconda3\lib\site-packages (from tensorflow-intel==2.18.
0->tensorflow) (3.12.1)
Requirement already satisfied: flatbuffers>=24.3.25 in c:\users\dell\anaconda3\lib\site-packages (from tensorflow-inte
l==2.18.0->tensorflow) (24.12.23)
Requirement already satisfied: keras>=3.5.0 in c:\users\dell\anaconda3\lib\site-packages (from tensorflow-intel==2.18.
0->tensorflow) (3.7.0)
Requirement already satisfied: packaging in c:\users\dell\anaconda3\lib\site-packages (from tensorflow-intel==2.18.0->
tensorflow) (22.0)
Requirement already satisfied: termcolor>=1.1.0 in c:\users\dell\anaconda3\lib\site-packages (from tensorflow-intel==
2.18.0->tensorflow) (2.5.0)
Requirement already satisfied: protobuf!=4.21.0,!=4.21.1,!=4.21.2,!=4.21.3,!=4.21.4,!=4.21.5,<6.0.0dev,>=3.20.3 in
c:\users\dell\anaconda3\lib\site-packages (from tensorflow-intel==2.18.0->tensorflow) (5.29.2)
Requirement already satisfied: libclang>=13.0.0 in c:\users\dell\anaconda3\lib\site-packages (from tensorflow-intel==
2.18.0->tensorflow) (18.1.1)
Requirement already satisfied: ml-dtypes<0.5.0,>=0.4.0 in c:\users\dell\anaconda3\lib\site-packages (from tensorflow-i
ntel==2.18.0->tensorflow) (0.4.1)
Requirement already satisfied: gast!=0.5.0,!=0.5.1,!=0.5.2,>=0.2.1 in c:\users\dell\anaconda3\lib\site-packages (from
tensorflow-intel==2.18.0->tensorflow) (0.6.0)
Requirement already satisfied: six>=1.12.0 in c:\users\dell\anaconda3\lib\site-packages (from tensorflow-intel==2.18.0
->tensorflow) (1.16.0)
Requirement already satisfied: absl-py>=1.0.0 in c:\users\dell\anaconda3\lib\site-packages (from tensorflow-intel==2.1
8.0->tensorflow) (2.1.0)
Requirement already satisfied: astunparse>=1.6.0 in c:\users\dell\anaconda3\lib\site-packages (from tensorflow-intel==
```

```
2.18.0->tensorflow) (1.6.3)
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in c:\users\dell\anaconda3\lib\site-packages (from
tensorflow-intel==2.18.0->tensorflow) (0.31.0)
Requirement already satisfied: google-pasta>=0.1.1 in c:\users\dell\anaconda3\lib\site-packages (from tensorflow-intel
==2.18.0->tensorflow) (0.2.0)
Requirement already satisfied: pytz>=2020.1 in c:\users\dell\anaconda3\lib\site-packages (from pandas) (2022.7)
Requirement already satisfied: python-dateutil>=2.8.1 in c:\users\dell\anaconda3\lib\site-packages (from pandas) (2.8.
2)
Requirement already satisfied: joblib>=1.1.1 in c:\users\dell\anaconda3\lib\site-packages (from scikit-learn) (1.1.1)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\dell\anaconda3\lib\site-packages (from scikit-learn)
(2.2.0)
Requirement already satisfied: scipy>=1.3.2 in c:\users\dell\anaconda3\lib\site-packages (from scikit-learn) (1.10.0)
Requirement already satisfied: wheel<1.0,>=0.23.0 in c:\users\dell\anaconda3\lib\site-packages (from astunparse>=1.6.0
->tensorflow-intel==2.18.0->tensorflow) (0.38.4)
Requirement already satisfied: namex in c:\users\dell\anaconda3\lib\site-packages (from keras>=3.5.0->tensorflow-intel
==2.18.0->tensorflow) (0.0.8)
Requirement already satisfied: optree in c:\users\dell\anaconda3\lib\site-packages (from keras>=3.5.0->tensorflow-inte
l==2.18.0->tensorflow) (0.13.1)
Requirement already satisfied: rich in c:\users\dell\anaconda3\lib\site-packages (from keras>=3.5.0->tensorflow-intel=
=2.18.0->tensorflow) (13.9.4)
Requirement already satisfied: charset-normalizer<3,>=2 in c:\users\dell\anaconda3\lib\site-packages (from requests<3,
>=2.21.0->tensorflow-intel==2.18.0->tensorflow) (2.0.4)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\dell\anaconda3\lib\site-packages (from requests<3,>=2.2
1.0->tensorflow-intel==2.18.0->tensorflow) (2023.5.7)
Requirement already satisfied: idna<4,>=2.5 in c:\users\dell\anaconda3\lib\site-packages (from requests<3,>=2.21.0->te
nsorflow-intel==2.18.0->tensorflow) (3.4)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\dell\anaconda3\lib\site-packages (from requests<3,>=
2.21.0->tensorflow-intel==2.18.0->tensorflow) (1.26.14)
Requirement already satisfied: tensorboard-data-server<0.8.0,>=0.7.0 in c:\users\dell\anaconda3\lib\site-packages (fro
m tensorboard<2.19,>=2.18->tensorflow-intel==2.18.0->tensorflow) (0.7.2)
Requirement already satisfied: werkzeug>=1.0.1 in c:\users\dell\anaconda3\lib\site-packages (from tensorboard<2.19,>=
2.18->tensorflow-intel==2.18.0->tensorflow) (2.2.2)
Requirement already satisfied: markdown>=2.6.8 in c:\users\dell\anaconda3\lib\site-packages (from tensorboard<2.19,>=
2.18->tensorflow-intel==2.18.0->tensorflow) (3.4.1)
Requirement already satisfied: MarkupSafe>=2.1.1 in c:\users\dell\anaconda3\lib\site-packages (from werkzeug>=1.0.1->t
ensorboard<2.19,>=2.18->tensorflow-intel==2.18.0->tensorflow) (2.1.1)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in c:\users\dell\anaconda3\lib\site-packages (from rich->keras>
=3.5.0->tensorflow-intel==2.18.0->tensorflow) (2.18.0)
Requirement already satisfied: markdown-it-py>=2.2.0 in c:\users\dell\anaconda3\lib\site-packages (from rich->keras>=
3.5.0->tensorflow-intel==2.18.0->tensorflow) (3.0.0)
Requirement already satisfied: mdurl~=0.1 in c:\users\dell\anaconda3\lib\site-packages (from markdown-it-py>=2.2.0->ri
ch->keras>=3.5.0->tensorflow-intel==2.18.0->tensorflow) (0.1.2)
Note: you may need to restart the kernel to use updated packages.
```

In [2]:
```
import numpy as np
import pandas as pd
import re
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras import layers, models
from sklearn.model_selection import train_test_split
```

In [3]:
```
data = {
    "input_texts": ["Hi", "How are you?", "What is your name?", "Bye", "Good morning",],
    "output_texts": ["Hello!", "I'm good, thank you!", "I'm a chatbot.", "Goodbye!", "Good morning!",]
}

df = pd.DataFrame(data)

print(df)
```

```
          input_texts          output_texts
0                  Hi                Hello!
1        How are you?  I'm good, thank you!
2  What is your name?        I'm a chatbot.
3                 Bye              Goodbye!
4        Good morning         Good morning!
```

In [4]:
```
def preprocess(texts):
    return [re.sub(r"[^a-zA-Z0-9\s]", "", text.lower()) for text in texts]

input_texts = preprocess(df['input_texts'])
output_texts = preprocess(df['output_texts'])

print("Processed Input Texts:", input_texts)
print("Processed Output Texts:", output_texts)
```

```
Processed Input Texts: ['hi', 'how are you', 'what is your name', 'bye', 'good morning']
Processed Output Texts: ['hello', 'im good thank you', 'im a chatbot', 'goodbye', 'good morning']
```

In [5]:
```
tokenizer_input = Tokenizer()
tokenizer_input.fit_on_texts(input_texts)
input_sequences = tokenizer_input.texts_to_sequences(input_texts)

tokenizer_output = Tokenizer()
tokenizer_output.fit_on_texts(output_texts)
output_sequences = tokenizer_output.texts_to_sequences(output_texts)

print("Input Sequences:", input_sequences)
print("Output Sequences:", output_sequences)
```

```
Input Sequences: [[1], [2, 3, 4], [5, 6, 7, 8], [9], [10, 11]]
Output Sequences: [[3], [1, 2, 4, 5], [1, 6, 7], [8], [2, 9]]
```

In [6]:
```
max_input_len = max(len(seq) for seq in input_sequences)
max_output_len = max(len(seq) for seq in output_sequences)

input_sequences = pad_sequences(input_sequences, maxlen=max_input_len, padding='post')
output_sequences = pad_sequences(output_sequences, maxlen=max_output_len, padding='post')

print("Padded Input Sequences:", input_sequences)
print("Padded Output Sequences:", output_sequences)
```

```
Padded Input Sequences: [[ 1  0  0  0]
 [ 2  3  4  0]
 [ 5  6  7  8]
 [ 9  0  0  0]
 [10 11  0  0]]
Padded Output Sequences: [[3 0 0 0]
 [1 2 4 5]
 [1 6 7 0]
 [8 0 0 0]
 [2 9 0 0]]
```

In [7]:
```
X_train, X_val, y_train, y_val = train_test_split(input_sequences, output_sequences, test_size=0.2)
print(f"Training data size: {X_train.shape[0]} | Validation data size: {X_val.shape[0]}")
```

```
Training data size: 4 | Validation data size: 1
```

In [8]:
```
def build_seq2seq_model(vocab_size_input, vocab_size_output, max_input_len, max_output_len):

    input_layer = layers.Input(shape=(max_input_len,))
    encoder_embedding = layers.Embedding(vocab_size_input, 128)(input_layer)
    encoder_lstm = layers.LSTM(128, return_state=True)
    encoder_output, state_h, state_c = encoder_lstm(encoder_embedding)

    decoder_input = layers.Input(shape=(max_output_len,))
    decoder_embedding = layers.Embedding(vocab_size_output, 128)(decoder_input)
    decoder_lstm = layers.LSTM(128, return_sequences=True, return_state=True)
    decoder_output, _, _ = decoder_lstm(decoder_embedding, initial_state=[state_h, state_c])

    output_layer = layers.Dense(vocab_size_output, activation='softmax')(decoder_output)

    model = models.Model([input_layer, decoder_input], output_layer)
    return model
```

In [9]:
```
model = build_seq2seq_model(
    vocab_size_input=len(tokenizer_input.word_index) + 1,
    vocab_size_output=len(tokenizer_output.word_index) + 1,
    max_input_len=max_input_len,
    max_output_len=max_output_len
)

model.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['accuracy'])
```

In [10]:
```
y_train = np.expand_dims(y_train, -1)
y_val = np.expand_dims(y_val, -1)
```

In [11]:
```
START_TOKEN = '<start>'
END_TOKEN = '<end>'

output_texts = [START_TOKEN + " " + text + " " + END_TOKEN for text in output_texts]

print("Modified Output Texts:", output_texts)
```

```
Modified Output Texts: ['<start> hello <end>', '<start> im good thank you <end>', '<start> im a chatbot <end>', '<star
t> goodbye <end>', '<start> good morning <end>']
```

In [12]:
```
tokenizer_input = Tokenizer()
tokenizer_input.fit_on_texts(input_texts)
input_sequences = tokenizer_input.texts_to_sequences(input_texts)

tokenizer_output = Tokenizer(filters='', lower=False)
tokenizer_output.fit_on_texts(output_texts)
output_sequences = tokenizer_output.texts_to_sequences(output_texts)

print("Input Sequences:", input_sequences)
print("Output Sequences:", output_sequences)
```

```
Input Sequences: [[1], [2, 3, 4], [5, 6, 7, 8], [9], [10, 11]]
Output Sequences: [[1, 5, 2], [1, 3, 4, 6, 7, 2], [1, 3, 8, 9, 2], [1, 10, 2], [1, 4, 11, 2]]
```

In [13]:
```
print("Input Tokenizer Word Index:", tokenizer_input.word_index)
print("Output Tokenizer Word Index:", tokenizer_output.word_index)
```

```
Input Tokenizer Word Index: {'hi': 1, 'how': 2, 'are': 3, 'you': 4, 'what': 5, 'is': 6, 'your': 7, 'name': 8, 'bye':
9, 'good': 10, 'morning': 11}
Output Tokenizer Word Index: {'<start>': 1, '<end>': 2, 'im': 3, 'good': 4, 'hello': 5, 'thank': 6, 'you': 7, 'a': 8,
'chatbot': 9, 'goodbye': 10, 'morning': 11}
```

In [14]:
```
def generate_response(input_text):

    input_seq = tokenizer_input.texts_to_sequences([input_text])
    input_seq = pad_sequences(input_seq, maxlen=max_input_len, padding='post')

    response_seq = [tokenizer_output.word_index[START_TOKEN]]

    for _ in range(max_output_len):
        pred = model.predict([input_seq, np.array([response_seq])])
        next_token = np.argmax(pred[0, -1, :])

        if next_token == tokenizer_output.word_index[END_TOKEN]:
            break

        response_seq.append(next_token)


    response_text = tokenizer_output.sequences_to_texts([response_seq])
    response_text = response_text[0].replace(START_TOKEN, '').replace(END_TOKEN, '').strip()

    return response_text
```

In [15]:
```
print(generate_response("Hi"))
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 1s/step

---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[15], line 1
----> 1 print(generate_response("Hi"))

Cell In[14], line 9, in generate_response(input_text)
      6 response_seq = [tokenizer_output.word_index[START_TOKEN]]
      8 for _ in range(max_output_len):
----> 9     pred = model.predict([input_seq, np.array([response_seq])])
     10     next_token = np.argmax(pred[0, -1, :])
     12     if next_token == tokenizer_output.word_index[END_TOKEN]:

File ~\anaconda3\lib\site-packages\keras\src\utils\traceback_utils.py:122, in filter_traceback.<locals>.error_handler
(*args, **kwargs)
    119     filtered_tb = _process_traceback_frames(e.__traceback__)
    120     # To get the full stack trace, call:
    121     # `keras.config.disable_traceback_filtering()`
--> 122     raise e.with_traceback(filtered_tb) from None
    123 finally:
    124     del filtered_tb

File ~\anaconda3\lib\site-packages\keras\src\layers\input_spec.py:245, in assert_input_compatibility(input_spec, input
s, layer_name)
    243 if spec_dim is not None and dim is not None:
    244     if spec_dim != dim:
--> 245         raise ValueError(
    246             f'Input {input_index} of layer "{layer_name}" is '
    247             "incompatible with the layer: "
    248             f"expected shape={spec.shape}, "
    249             f"found shape={shape}"
    250         )

ValueError: Input 1 of layer "functional" is incompatible with the layer: expected shape=(None, 4), found shape=(1, 2)
```

In [ ]:

In [ ]: