

Autocompletion

May 3, 2024

1 Import Necessary Libraries

```
[1]: import json
import numpy as np
import tensorflow as tf
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Embedding, LSTM, Dense, Bidirectional
from tensorflow.keras.optimizers import Adam
```

WARNING:tensorflow:From C:\Users\Advait Amit
Kisar\AppData\Local\Programs\Python\Python311\Lib\site-
packages\keras\src\losses.py:2976: The name
tf.losses.sparse_softmax_cross_entropy is deprecated. Please use
tf.compat.v1.losses.sparse_softmax_cross_entropy instead.

```
[2]: def load_queries_from_json(file_path):
    """Load queries from a JSON file."""
    queries_json = json.load(open(file_path, 'r'))
    query_ids = [item["query number"] for item in queries_json]
    queries = [item["query"] for item in queries_json]
    return query_ids, queries
```

```
[3]: def preprocess_queries(queries):
    """Preprocess queries by merging and splitting into sentences."""
    query_merged = ' '.join(queries)
    queries_sent = query_merged.split('.')
    return queries_sent
```

```
[4]: def tokenize_queries(queries_complete):
    """Tokenize queries."""
    tokenizer = Tokenizer()
    tokenizer.fit_on_texts(queries_complete)
    return tokenizer
```

```

[5]: def generate_input_sequences(tokenizer, queries_complete):
    """Generate input sequences for training."""
    input_sequences = []
    for query in queries_complete:
        sequence = tokenizer.texts_to_sequences([query])[0]
        for i in range(1, len(sequence)):
            input_sequences.append(sequence[:i+1])
    return input_sequences

[6]: def preprocess_input_sequences(input_sequences, max_seq_len):
    """Pad input sequences to equal length."""
    input_sequences = np.array(pad_sequences(input_sequences,
    ↪maxlen=max_seq_len, padding='pre'))
    return input_sequences

[7]: def create_model_and_train(x_sequences, y_sequences, total_words, max_seq_len,
    ↪n_epochs=200):
    """Create and train the LSTM model."""
    model = Sequential()
    model.add(Embedding(total_words, 16, input_length=max_seq_len - 1))
    model.add(Bidirectional(LSTM(50)))
    model.add(Dense(total_words, activation='softmax'))
    model.compile(loss='categorical_crossentropy', optimizer='adam',
    ↪metrics=['accuracy'])
    model.summary()
    history = model.fit(x_sequences, y_sequences, epochs=n_epochs, verbose=1) #
    ↪try to run for full 200 epochs atleast for some meaningful completion
    return model

[8]: def complete_query(model, tokenizer, reverse_word_index, incomplete_query,
    ↪next_n_words=1, max_seq_len=0):
    """Complete the given query by predicting the next n words."""
    seed_text = incomplete_query

    for i in range(next_n_words):
        token_list = tokenizer.texts_to_sequences([seed_text])[0] # list of
        ↪list
        token_list = pad_sequences([token_list], maxlen=max_seq_len - 1,
        ↪padding='pre')

        # Use model.predict to get probabilities for all words
        predictions = model.predict(token_list, verbose=0)[0]

        # Use argmax to get the index of the word with highest probability
        predicted_index = np.argmax(predictions)
        out_word = reverse_word_index[predicted_index]
        seed_text += " " + out_word

```

```
return seed_text
```

```
[9]: # Load queries
query_ids, queries = load_queries_from_json("cran_queries.json")
print("Number of queries:", len(queries))
```

Number of queries: 225

```
[10]: # Preprocess queries
sentences = preprocess_queries(queries)
```

```
[11]: # Tokenize queries
tokenizer = tokenize_queries(sentences)
total_words = len(tokenizer.word_index) + 1
```

```
[12]: # Generate input sequences
input_sequences = generate_input_sequences(tokenizer, sentences)
```

```
[13]: # Pad input sequences
max_sequence_length = max([len(sequence) for sequence in input_sequences])
padded_sequences = preprocess_input_sequences(input_sequences,
↪max_sequence_length)
```

```
[14]: # Create predictors and labels
predictors, labels = padded_sequences[:, :-1], padded_sequences[:, -1]
one_hot_labels = tf.keras.utils.to_categorical(labels, num_classes=total_words)
```

```
[15]: trained_model = create_model_and_train(predictors, one_hot_labels, total_words,
↪max_sequence_length, n_epochs=200)
```

WARNING:tensorflow:From C:\Users\Advait Amit
Kisar\AppData\Local\Programs\Python\Python311\Lib\site-
packages\keras\src\backend.py:873: The name tf.get_default_graph is deprecated.
Please use tf.compat.v1.get_default_graph instead.

WARNING:tensorflow:From C:\Users\Advait Amit
Kisar\AppData\Local\Programs\Python\Python311\Lib\site-
packages\keras\src\optimizers_init_.py:309: The name tf.train.Optimizer is
deprecated. Please use tf.compat.v1.train.Optimizer instead.

Model: "sequential"

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 49, 16)	15296
bidirectional (Bidirectional)	(None, 100)	26800

dense (Dense) (None, 956) 96556

=====

Total params: 138652 (541.61 KB)

Trainable params: 138652 (541.61 KB)

Non-trainable params: 0 (0.00 Byte)

Epoch 1/200

WARNING:tensorflow:From C:\Users\Advait Amit

Kisar\AppData\Local\Programs\Python\Python311\Lib\site-

packages\keras\src\utils\tf_utils.py:492: The name tf.ragged.RaggedTensorValue
is deprecated. Please use tf.compat.v1.ragged.RaggedTensorValue instead.

WARNING:tensorflow:From C:\Users\Advait Amit

Kisar\AppData\Local\Programs\Python\Python311\Lib\site-

packages\keras\src\engine\base_layer_utils.py:384: The name
tf.executing_eagerly_outside_functions is deprecated. Please use
tf.compat.v1.executing_eagerly_outside_functions instead.

115/115 [=====] - 18s 44ms/step - loss: 6.2713 -
accuracy: 0.0625

Epoch 2/200

115/115 [=====] - 5s 45ms/step - loss: 5.8476 -
accuracy: 0.0634

Epoch 3/200

115/115 [=====] - 5s 43ms/step - loss: 5.7770 -
accuracy: 0.0677

Epoch 4/200

115/115 [=====] - 5s 45ms/step - loss: 5.6939 -
accuracy: 0.0718

Epoch 5/200

115/115 [=====] - 5s 45ms/step - loss: 5.5925 -
accuracy: 0.0822

Epoch 6/200

115/115 [=====] - 5s 42ms/step - loss: 5.4826 -
accuracy: 0.0972

Epoch 7/200

115/115 [=====] - 5s 43ms/step - loss: 5.3746 -
accuracy: 0.1035

Epoch 8/200

115/115 [=====] - 5s 46ms/step - loss: 5.2754 -
accuracy: 0.1139

Epoch 9/200

115/115 [=====] - 5s 45ms/step - loss: 5.1894 -
accuracy: 0.1202

Epoch 10/200

115/115 [=====] - 5s 45ms/step - loss: 5.1076 -

```

accuracy: 0.1264
Epoch 11/200
115/115 [=====] - 5s 45ms/step - loss: 5.0280 -
accuracy: 0.1270
Epoch 12/200
115/115 [=====] - 5s 45ms/step - loss: 4.9502 -
accuracy: 0.1311
Epoch 13/200
115/115 [=====] - 5s 44ms/step - loss: 4.8754 -
accuracy: 0.1352
Epoch 14/200
115/115 [=====] - 5s 45ms/step - loss: 4.8017 -
accuracy: 0.1401
Epoch 15/200
115/115 [=====] - 5s 43ms/step - loss: 4.7298 -
accuracy: 0.1442
Epoch 16/200
115/115 [=====] - 5s 44ms/step - loss: 4.6553 -
accuracy: 0.1483
Epoch 17/200
115/115 [=====] - 5s 45ms/step - loss: 4.5825 -
accuracy: 0.1592
Epoch 18/200
115/115 [=====] - 5s 43ms/step - loss: 4.5125 -
accuracy: 0.1614
Epoch 19/200
115/115 [=====] - 5s 44ms/step - loss: 4.4438 -
accuracy: 0.1690
Epoch 20/200
115/115 [=====] - 5s 43ms/step - loss: 4.3716 -
accuracy: 0.1780
Epoch 21/200
115/115 [=====] - 5s 44ms/step - loss: 4.3331 -
accuracy: 0.1761
Epoch 22/200
115/115 [=====] - 5s 43ms/step - loss: 4.2429 -
accuracy: 0.1860
Epoch 23/200
115/115 [=====] - 5s 45ms/step - loss: 4.1661 -
accuracy: 0.1901
Epoch 24/200
115/115 [=====] - 5s 44ms/step - loss: 4.0938 -
accuracy: 0.1947
Epoch 25/200
115/115 [=====] - 5s 45ms/step - loss: 4.0226 -
accuracy: 0.2056
Epoch 26/200
115/115 [=====] - 5s 45ms/step - loss: 3.9540 -

```

```

accuracy: 0.2094
Epoch 27/200
115/115 [=====] - 5s 44ms/step - loss: 3.8887 -
accuracy: 0.2209
Epoch 28/200
115/115 [=====] - 5s 46ms/step - loss: 3.8196 -
accuracy: 0.2272
Epoch 29/200
115/115 [=====] - 5s 42ms/step - loss: 3.7521 -
accuracy: 0.2332
Epoch 30/200
115/115 [=====] - 5s 42ms/step - loss: 3.6866 -
accuracy: 0.2373
Epoch 31/200
115/115 [=====] - 5s 45ms/step - loss: 3.6217 -
accuracy: 0.2499
Epoch 32/200
115/115 [=====] - 5s 44ms/step - loss: 3.5557 -
accuracy: 0.2581
Epoch 33/200
115/115 [=====] - 5s 44ms/step - loss: 3.4866 -
accuracy: 0.2632
Epoch 34/200
115/115 [=====] - 5s 42ms/step - loss: 3.4227 -
accuracy: 0.2783
Epoch 35/200
115/115 [=====] - 5s 43ms/step - loss: 3.3638 -
accuracy: 0.2840
Epoch 36/200
115/115 [=====] - 5s 44ms/step - loss: 3.2983 -
accuracy: 0.2982
Epoch 37/200
115/115 [=====] - 5s 43ms/step - loss: 3.2378 -
accuracy: 0.3058
Epoch 38/200
115/115 [=====] - 5s 46ms/step - loss: 3.1836 -
accuracy: 0.3192
Epoch 39/200
115/115 [=====] - 5s 44ms/step - loss: 3.1219 -
accuracy: 0.3258
Epoch 40/200
115/115 [=====] - 5s 44ms/step - loss: 3.0599 -
accuracy: 0.3394
Epoch 41/200
115/115 [=====] - 5s 44ms/step - loss: 2.9986 -
accuracy: 0.3520
Epoch 42/200
115/115 [=====] - 5s 43ms/step - loss: 2.9390 -

```

accuracy: 0.3640
Epoch 43/200
115/115 [=====] - 5s 44ms/step - loss: 2.8831 -
accuracy: 0.3747
Epoch 44/200
115/115 [=====] - 6s 50ms/step - loss: 2.8238 -
accuracy: 0.3848
Epoch 45/200
115/115 [=====] - 5s 43ms/step - loss: 2.7703 -
accuracy: 0.3960
Epoch 46/200
115/115 [=====] - 5s 43ms/step - loss: 2.7141 -
accuracy: 0.4137
Epoch 47/200
115/115 [=====] - 5s 42ms/step - loss: 2.6586 -
accuracy: 0.4205
Epoch 48/200
115/115 [=====] - 5s 42ms/step - loss: 2.6027 -
accuracy: 0.4369
Epoch 49/200
115/115 [=====] - 5s 42ms/step - loss: 2.5509 -
accuracy: 0.4487
Epoch 50/200
115/115 [=====] - 5s 43ms/step - loss: 2.4913 -
accuracy: 0.4656
Epoch 51/200
115/115 [=====] - 5s 42ms/step - loss: 2.4374 -
accuracy: 0.4724
Epoch 52/200
115/115 [=====] - 5s 43ms/step - loss: 2.3845 -
accuracy: 0.4831
Epoch 53/200
115/115 [=====] - 5s 44ms/step - loss: 2.3333 -
accuracy: 0.4973
Epoch 54/200
115/115 [=====] - 5s 41ms/step - loss: 2.2840 -
accuracy: 0.5038
Epoch 55/200
115/115 [=====] - 5s 42ms/step - loss: 2.2340 -
accuracy: 0.5188
Epoch 56/200
115/115 [=====] - 5s 41ms/step - loss: 2.1808 -
accuracy: 0.5328
Epoch 57/200
115/115 [=====] - 5s 40ms/step - loss: 2.1345 -
accuracy: 0.5415
Epoch 58/200
115/115 [=====] - 5s 41ms/step - loss: 2.0874 -

accuracy: 0.5546
Epoch 59/200
115/115 [=====] - 5s 42ms/step - loss: 2.0452 -
accuracy: 0.5601
Epoch 60/200
115/115 [=====] - 5s 41ms/step - loss: 1.9950 -
accuracy: 0.5699
Epoch 61/200
115/115 [=====] - 5s 43ms/step - loss: 1.9506 -
accuracy: 0.5844
Epoch 62/200
115/115 [=====] - 5s 41ms/step - loss: 1.9077 -
accuracy: 0.5950
Epoch 63/200
115/115 [=====] - 5s 41ms/step - loss: 1.8715 -
accuracy: 0.6010
Epoch 64/200
115/115 [=====] - 5s 41ms/step - loss: 1.8234 -
accuracy: 0.6111
Epoch 65/200
115/115 [=====] - 5s 41ms/step - loss: 1.7842 -
accuracy: 0.6237
Epoch 66/200
115/115 [=====] - 5s 42ms/step - loss: 1.7444 -
accuracy: 0.6346
Epoch 67/200
115/115 [=====] - 5s 41ms/step - loss: 1.7163 -
accuracy: 0.6379
Epoch 68/200
115/115 [=====] - 5s 41ms/step - loss: 1.6703 -
accuracy: 0.6450
Epoch 69/200
115/115 [=====] - 5s 41ms/step - loss: 1.6309 -
accuracy: 0.6546
Epoch 70/200
115/115 [=====] - 5s 40ms/step - loss: 1.5875 -
accuracy: 0.6644
Epoch 71/200
115/115 [=====] - 5s 41ms/step - loss: 1.5525 -
accuracy: 0.6734
Epoch 72/200
115/115 [=====] - 5s 40ms/step - loss: 1.5198 -
accuracy: 0.6808
Epoch 73/200
115/115 [=====] - 5s 41ms/step - loss: 1.4874 -
accuracy: 0.6920
Epoch 74/200
115/115 [=====] - 5s 40ms/step - loss: 1.4551 -


```

accuracy: 0.7054
Epoch 75/200
115/115 [=====] - 5s 41ms/step - loss: 1.4151 -
accuracy: 0.7103
Epoch 76/200
115/115 [=====] - 5s 42ms/step - loss: 1.3902 -
accuracy: 0.7130
Epoch 77/200
115/115 [=====] - 5s 41ms/step - loss: 1.3731 -
accuracy: 0.7163
Epoch 78/200
115/115 [=====] - 5s 41ms/step - loss: 1.3293 -
accuracy: 0.7239
Epoch 79/200
115/115 [=====] - 5s 41ms/step - loss: 1.3032 -
accuracy: 0.7365
Epoch 80/200
115/115 [=====] - 5s 41ms/step - loss: 1.2658 -
accuracy: 0.7428
Epoch 81/200
115/115 [=====] - 5s 41ms/step - loss: 1.2397 -
accuracy: 0.7556
Epoch 82/200
115/115 [=====] - 5s 41ms/step - loss: 1.2210 -
accuracy: 0.7526
Epoch 83/200
115/115 [=====] - 5s 40ms/step - loss: 1.1814 -
accuracy: 0.7638
Epoch 84/200
115/115 [=====] - 5s 41ms/step - loss: 1.1664 -
accuracy: 0.7643
Epoch 85/200
115/115 [=====] - 5s 41ms/step - loss: 1.1311 -
accuracy: 0.7728
Epoch 86/200
115/115 [=====] - 5s 40ms/step - loss: 1.1095 -
accuracy: 0.7799
Epoch 87/200
115/115 [=====] - 5s 40ms/step - loss: 1.0900 -
accuracy: 0.7835
Epoch 88/200
115/115 [=====] - 5s 42ms/step - loss: 1.0655 -
accuracy: 0.7862
Epoch 89/200
115/115 [=====] - 5s 40ms/step - loss: 1.0333 -
accuracy: 0.7946
Epoch 90/200
115/115 [=====] - 5s 41ms/step - loss: 1.0175 -

```

```

accuracy: 0.7982
Epoch 91/200
115/115 [=====] - 5s 41ms/step - loss: 1.0013 -
accuracy: 0.8039
Epoch 92/200
115/115 [=====] - 5s 40ms/step - loss: 0.9833 -
accuracy: 0.8039
Epoch 93/200
115/115 [=====] - 5s 42ms/step - loss: 0.9585 -
accuracy: 0.8075
Epoch 94/200
115/115 [=====] - 5s 41ms/step - loss: 0.9342 -
accuracy: 0.8159
Epoch 95/200
115/115 [=====] - 5s 41ms/step - loss: 0.9286 -
accuracy: 0.8211
Epoch 96/200
115/115 [=====] - 5s 40ms/step - loss: 0.8958 -
accuracy: 0.8217
Epoch 97/200
115/115 [=====] - 5s 41ms/step - loss: 0.8777 -
accuracy: 0.8326
Epoch 98/200
115/115 [=====] - 5s 47ms/step - loss: 0.8828 -
accuracy: 0.8271
Epoch 99/200
115/115 [=====] - 5s 45ms/step - loss: 0.8455 -
accuracy: 0.8340
Epoch 100/200
115/115 [=====] - 5s 45ms/step - loss: 0.8305 -
accuracy: 0.8359
Epoch 101/200
115/115 [=====] - 5s 44ms/step - loss: 0.8105 -
accuracy: 0.8416
Epoch 102/200
115/115 [=====] - 5s 43ms/step - loss: 0.8156 -
accuracy: 0.8381
Epoch 103/200
115/115 [=====] - 5s 43ms/step - loss: 0.7875 -
accuracy: 0.8468
Epoch 104/200
115/115 [=====] - 5s 42ms/step - loss: 0.7726 -
accuracy: 0.8465
Epoch 105/200
115/115 [=====] - 5s 42ms/step - loss: 0.7516 -
accuracy: 0.8504
Epoch 106/200
115/115 [=====] - 5s 42ms/step - loss: 0.7413 -

```

```

accuracy: 0.8572
Epoch 107/200
115/115 [=====] - 5s 43ms/step - loss: 0.7156 -
accuracy: 0.8626
Epoch 108/200
115/115 [=====] - 5s 41ms/step - loss: 0.7031 -
accuracy: 0.8643
Epoch 109/200
115/115 [=====] - 5s 41ms/step - loss: 0.7037 -
accuracy: 0.8607
Epoch 110/200
115/115 [=====] - 5s 41ms/step - loss: 0.6958 -
accuracy: 0.8667
Epoch 111/200
115/115 [=====] - 5s 40ms/step - loss: 0.6765 -
accuracy: 0.8689
Epoch 112/200
115/115 [=====] - 5s 41ms/step - loss: 0.6534 -
accuracy: 0.8758
Epoch 113/200
115/115 [=====] - 5s 42ms/step - loss: 0.6515 -
accuracy: 0.8733
Epoch 114/200
115/115 [=====] - 5s 40ms/step - loss: 0.6386 -
accuracy: 0.8760
Epoch 115/200
115/115 [=====] - 5s 41ms/step - loss: 0.6310 -
accuracy: 0.8807
Epoch 116/200
115/115 [=====] - 5s 41ms/step - loss: 0.6160 -
accuracy: 0.8807
Epoch 117/200
115/115 [=====] - 5s 41ms/step - loss: 0.6091 -
accuracy: 0.8837
Epoch 118/200
115/115 [=====] - 5s 41ms/step - loss: 0.5966 -
accuracy: 0.8829
Epoch 119/200
115/115 [=====] - 5s 41ms/step - loss: 0.5858 -
accuracy: 0.8831
Epoch 120/200
115/115 [=====] - 5s 41ms/step - loss: 0.5795 -
accuracy: 0.8883
Epoch 121/200
115/115 [=====] - 5s 41ms/step - loss: 0.5764 -
accuracy: 0.8894
Epoch 122/200
115/115 [=====] - 5s 41ms/step - loss: 0.5619 -

```

accuracy: 0.8853
Epoch 123/200
115/115 [=====] - 5s 41ms/step - loss: 0.5436 -
accuracy: 0.8938
Epoch 124/200
115/115 [=====] - 5s 41ms/step - loss: 0.5433 -
accuracy: 0.8919
Epoch 125/200
115/115 [=====] - 5s 40ms/step - loss: 0.5262 -
accuracy: 0.8995
Epoch 126/200
115/115 [=====] - 5s 42ms/step - loss: 0.5179 -
accuracy: 0.8968
Epoch 127/200
115/115 [=====] - 5s 41ms/step - loss: 0.5081 -
accuracy: 0.9003
Epoch 128/200
115/115 [=====] - 5s 41ms/step - loss: 0.5000 -
accuracy: 0.9022
Epoch 129/200
115/115 [=====] - 5s 41ms/step - loss: 0.4982 -
accuracy: 0.8995
Epoch 130/200
115/115 [=====] - 5s 41ms/step - loss: 0.5308 -
accuracy: 0.8891
Epoch 131/200
115/115 [=====] - 5s 41ms/step - loss: 0.4964 -
accuracy: 0.9001
Epoch 132/200
115/115 [=====] - 5s 41ms/step - loss: 0.4765 -
accuracy: 0.9017
Epoch 133/200
115/115 [=====] - 5s 40ms/step - loss: 0.4620 -
accuracy: 0.9082
Epoch 134/200
115/115 [=====] - 5s 40ms/step - loss: 0.4532 -
accuracy: 0.9066
Epoch 135/200
115/115 [=====] - 5s 41ms/step - loss: 0.4461 -
accuracy: 0.9088
Epoch 136/200
115/115 [=====] - 5s 41ms/step - loss: 0.4471 -
accuracy: 0.9077
Epoch 137/200
115/115 [=====] - 5s 41ms/step - loss: 0.4494 -
accuracy: 0.9088
Epoch 138/200
115/115 [=====] - 5s 41ms/step - loss: 0.4555 -

```

accuracy: 0.9058
Epoch 139/200
115/115 [=====] - 5s 41ms/step - loss: 0.4320 -
accuracy: 0.9123
Epoch 140/200
115/115 [=====] - 5s 40ms/step - loss: 0.4569 -
accuracy: 0.9052
Epoch 141/200
115/115 [=====] - 5s 41ms/step - loss: 0.4401 -
accuracy: 0.9077
Epoch 142/200
115/115 [=====] - 5s 40ms/step - loss: 0.4233 -
accuracy: 0.9123
Epoch 143/200
115/115 [=====] - 5s 41ms/step - loss: 0.4091 -
accuracy: 0.9145
Epoch 144/200
115/115 [=====] - 5s 41ms/step - loss: 0.4053 -
accuracy: 0.9153
Epoch 145/200
115/115 [=====] - 5s 41ms/step - loss: 0.3936 -
accuracy: 0.9200
Epoch 146/200
115/115 [=====] - 5s 40ms/step - loss: 0.3889 -
accuracy: 0.9153
Epoch 147/200
115/115 [=====] - 5s 41ms/step - loss: 0.3827 -
accuracy: 0.9170
Epoch 148/200
115/115 [=====] - 5s 41ms/step - loss: 0.3809 -
accuracy: 0.9143
Epoch 149/200
115/115 [=====] - 5s 41ms/step - loss: 0.3795 -
accuracy: 0.9192
Epoch 150/200
115/115 [=====] - 5s 40ms/step - loss: 0.3742 -
accuracy: 0.9178
Epoch 151/200
115/115 [=====] - 5s 40ms/step - loss: 0.3686 -
accuracy: 0.9194
Epoch 152/200
115/115 [=====] - 5s 41ms/step - loss: 0.3673 -
accuracy: 0.9194
Epoch 153/200
115/115 [=====] - 5s 41ms/step - loss: 0.3563 -
accuracy: 0.9178
Epoch 154/200
115/115 [=====] - 5s 41ms/step - loss: 0.3789 -

```

```

accuracy: 0.9143
Epoch 155/200
115/115 [=====] - 5s 41ms/step - loss: 0.3707 -
accuracy: 0.9159
Epoch 156/200
115/115 [=====] - 5s 41ms/step - loss: 0.3817 -
accuracy: 0.9113
Epoch 157/200
115/115 [=====] - 5s 41ms/step - loss: 0.3585 -
accuracy: 0.9173
Epoch 158/200
115/115 [=====] - 5s 41ms/step - loss: 0.3430 -
accuracy: 0.9222
Epoch 159/200
115/115 [=====] - 5s 41ms/step - loss: 0.3371 -
accuracy: 0.9227
Epoch 160/200
115/115 [=====] - 5s 41ms/step - loss: 0.3389 -
accuracy: 0.9214
Epoch 161/200
115/115 [=====] - 5s 41ms/step - loss: 0.3428 -
accuracy: 0.9208
Epoch 162/200
115/115 [=====] - 5s 41ms/step - loss: 0.3325 -
accuracy: 0.9219
Epoch 163/200
115/115 [=====] - 5s 41ms/step - loss: 0.3291 -
accuracy: 0.9227
Epoch 164/200
115/115 [=====] - 5s 41ms/step - loss: 0.3243 -
accuracy: 0.9211
Epoch 165/200
115/115 [=====] - 5s 41ms/step - loss: 0.3239 -
accuracy: 0.9205
Epoch 166/200
115/115 [=====] - 5s 40ms/step - loss: 0.3246 -
accuracy: 0.9219
Epoch 167/200
115/115 [=====] - 5s 41ms/step - loss: 0.3124 -
accuracy: 0.9257
Epoch 168/200
115/115 [=====] - 5s 41ms/step - loss: 0.3086 -
accuracy: 0.9208
Epoch 169/200
115/115 [=====] - 5s 41ms/step - loss: 0.3147 -
accuracy: 0.9233
Epoch 170/200
115/115 [=====] - 5s 41ms/step - loss: 0.3186 -

```

accuracy: 0.9224
Epoch 171/200
115/115 [=====] - 5s 41ms/step - loss: 0.3052 -
accuracy: 0.9257
Epoch 172/200
115/115 [=====] - 5s 41ms/step - loss: 0.3006 -
accuracy: 0.9238
Epoch 173/200
115/115 [=====] - 5s 41ms/step - loss: 0.3277 -
accuracy: 0.9189
Epoch 174/200
115/115 [=====] - 5s 41ms/step - loss: 0.3278 -
accuracy: 0.9219
Epoch 175/200
115/115 [=====] - 5s 42ms/step - loss: 0.3047 -
accuracy: 0.9235
Epoch 176/200
115/115 [=====] - 5s 41ms/step - loss: 0.2964 -
accuracy: 0.9216
Epoch 177/200
115/115 [=====] - 5s 41ms/step - loss: 0.2954 -
accuracy: 0.9249
Epoch 178/200
115/115 [=====] - 5s 41ms/step - loss: 0.2866 -
accuracy: 0.9233
Epoch 179/200
115/115 [=====] - 5s 40ms/step - loss: 0.2897 -
accuracy: 0.9230
Epoch 180/200
115/115 [=====] - 5s 42ms/step - loss: 0.2859 -
accuracy: 0.9235
Epoch 181/200
115/115 [=====] - 5s 42ms/step - loss: 0.2802 -
accuracy: 0.9233
Epoch 182/200
115/115 [=====] - 5s 41ms/step - loss: 0.2759 -
accuracy: 0.9235
Epoch 183/200
115/115 [=====] - 5s 41ms/step - loss: 0.2742 -
accuracy: 0.9249
Epoch 184/200
115/115 [=====] - 5s 41ms/step - loss: 0.2762 -
accuracy: 0.9235
Epoch 185/200
115/115 [=====] - 5s 41ms/step - loss: 0.2774 -
accuracy: 0.9279
Epoch 186/200
115/115 [=====] - 5s 41ms/step - loss: 0.2759 -

```

accuracy: 0.9224
Epoch 187/200
115/115 [=====] - 5s 40ms/step - loss: 0.2693 -
accuracy: 0.9252
Epoch 188/200
115/115 [=====] - 5s 40ms/step - loss: 0.2682 -
accuracy: 0.9244
Epoch 189/200
115/115 [=====] - 5s 40ms/step - loss: 0.2681 -
accuracy: 0.9260
Epoch 190/200
115/115 [=====] - 5s 41ms/step - loss: 0.2692 -
accuracy: 0.9235
Epoch 191/200
115/115 [=====] - 5s 41ms/step - loss: 0.3025 -
accuracy: 0.9170
Epoch 192/200
115/115 [=====] - 5s 41ms/step - loss: 0.2957 -
accuracy: 0.9230
Epoch 193/200
115/115 [=====] - 5s 41ms/step - loss: 0.2758 -
accuracy: 0.9235
Epoch 194/200
115/115 [=====] - 5s 42ms/step - loss: 0.2638 -
accuracy: 0.9263
Epoch 195/200
115/115 [=====] - 5s 42ms/step - loss: 0.2587 -
accuracy: 0.9255
Epoch 196/200
115/115 [=====] - 5s 41ms/step - loss: 0.2579 -
accuracy: 0.9246
Epoch 197/200
115/115 [=====] - 5s 41ms/step - loss: 0.2541 -
accuracy: 0.9265
Epoch 198/200
115/115 [=====] - 5s 40ms/step - loss: 0.2560 -
accuracy: 0.9263
Epoch 199/200
115/115 [=====] - 5s 41ms/step - loss: 0.2608 -
accuracy: 0.9249
Epoch 200/200
115/115 [=====] - 5s 41ms/step - loss: 0.2620 -
accuracy: 0.9260

```

```

[16]: # Reverse word index
reverse_word_index = {index: word for (word, index) in tokenizer.word_index.
    ↪items()}

```



```
[17]: # Queries
original_queries = [
    "to find an approximate correction for thickness in slender thin-wing
    ↳theory .",
    "why does the compressibility transformation fail to correlate the high
    ↳speed data for helium and air .",
    "how is the heat transfer downstream of the mass transfer region effected
    ↳by mass transfer at the nose of a blunted cone .",
    "does transition in the hypersonic wake depend on body geometry and size",
    "what is a criterion that the transonic flow around an airfoil with a round
    ↳leading edge be validly analyzed by the linearized transonic flow theory ."
]

incomplete_queries = [
    "to find an approximate correction for",
    "why does the compressibility transformation fail to correlate the high",
    "how is the heat transfer downstream of the mass transfer region effected
    ↳by mass transfer at",
    "does transition in the hypersonic wake",
    "what is a criterion that the transonic flow around an airfoil with a round
    ↳leading edge be validly analyzed"
]
```

```
[18]: # Complete queries
for idx, incomplete_query in enumerate(incomplete_queries):
    print(f"Processing Query No. {idx+1}:")
    completed_query_n_words = complete_query(trained_model, tokenizer,
    ↳reverse_word_index, incomplete_query, next_n_words=6,
    ↳max_seq_len=max_sequence_length)
    print("Incomplete query:", incomplete_query)
    print("Predicted Complete query:", completed_query_n_words)
    print("Original Complete query:", original_queries[idx], "\n")
```

Processing Query No. 1:

Incomplete query: to find an approximate correction for

Predicted Complete query: to find an approximate correction for thickness in slender thin wing theory

Original Complete query: to find an approximate correction for thickness in slender thin-wing theory .

Processing Query No. 2:

Incomplete query: why does the compressibility transformation fail to correlate the high

Predicted Complete query: why does the compressibility transformation fail to correlate the high speed data for helium and air

Original Complete query: why does the compressibility transformation fail to correlate the high speed data for helium and air .

Processing Query No. 3:

Incomplete query: how is the heat transfer downstream of the mass transfer region effected by mass transfer at

Predicted Complete query: how is the heat transfer downstream of the mass transfer region effected by mass transfer at the nose of a blunted cone

Original Complete query: how is the heat transfer downstream of the mass transfer region effected by mass transfer at the nose of a blunted cone .

Processing Query No. 4:

Incomplete query: does transition in the hypersonic wake

Predicted Complete query: does transition in the hypersonic wake depend on body geometry and size

Original Complete query: does transition in the hypersonic wake depend on body geometry and size

Processing Query No. 5:

Incomplete query: what is a criterion that the transonic flow around an airfoil with a round leading edge be validly analyzed

Predicted Complete query: what is a criterion that the transonic flow around an airfoil with a round leading edge be validly analyzed by the linearized transonic flow theory

Original Complete query: what is a criterion that the transonic flow around an airfoil with a round leading edge be validly analyzed by the linearized transonic flow theory .