```python
import pandas as pd
import numpy as np
import nltk
import json
from tensorflow import keras
import tensorflow as tf

def read_jsonl(filename):
    data = []
    with open(filename, 'r') as file:
        for line in file:
            data.append(json.loads(line))
    return data

filename = 'train.jsonl'
jsonl_data = read_jsonl(filename)

df = pd.DataFrame(jsonl_data)

df
```

```
                             id  \
0             gigaword-train-0
1             gigaword-train-1
2             gigaword-train-2
3             gigaword-train-3
4             gigaword-train-4
...                        ...
999995  gigaword-train-999995
999996  gigaword-train-999996
999997  gigaword-train-999997
999998  gigaword-train-999998
999999  gigaword-train-999999

                                                     text  \
0       australia 's current account deficit shrunk by...
1       at least two people were killed in a suspected...
2       australian shares closed down #.# percent mond...
3       south korea 's nuclear envoy kim sook urged no...
4       south korea on monday announced sweeping tax r...
...                                                   ...
999995  after proclaiming a special relationship with ...
999996  a group of people expelled by the british from...
999997  a mix of profit-taking and cautiousness guided...
999998  hungary 's air carrier , malev , has grounded ...
999999  a ##-year-old-girl who struck prince charles i...

                                                  summary
0       australian current account deficit narrows sha...
1         at least two dead in southern philippines blast
```

```
2                 australian stocks close down #.# percent
3         envoy urges north korea to restart nuclear dis...
4            skorea announces tax cuts to stimulate economy
...                                                     ...
999995  indian leader vajpayee to meet with bush to di...
999996  former residents of indian ocean island demand...
999997                       stocks lower in early trading
999998    hungarian air carrier grounds flights to bosnia
999999  teen-ager who struck prince charles with carna...

[1000000 rows x 3 columns]
```

```
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Embedding, SimpleRNN, GRU, LSTM,
Dense

tokenizer = Tokenizer()
tokenizer.fit_on_texts(df['text'])
sequences = tokenizer.texts_to_sequences(df['text'])
word_index = tokenizer.word_index

summary_sequences = tokenizer.texts_to_sequences(df['summary'])

maxlen = 15
data_pad = pad_sequences(sequences, maxlen=maxlen)

summary_pad = pad_sequences(summary_sequences, maxlen=maxlen)

embedding_dim = 50
vocab_size = len(word_index) + 1

vocab_size
```

```
78846
```

```
model_rnn = Sequential([
    Embedding(maxlen, embedding_dim),
    SimpleRNN(32, return_sequences=True),
    Dense(vocab_size, activation='softmax')
])
model_rnn.compile(optimizer='adam',
loss='sparse_categorical_crossentropy')

len(word_index)
```

```
78845
```

```
model_rnn.fit(data_pad, summary_pad, epochs=20, batch_size=32)
```

```
Epoch 1/20
31250/31250 [==============================] - 1270s 41ms/step - loss:
4.1973
Epoch 2/20
31250/31250 [==============================] - 1306s 42ms/step - loss:
4.1933
Epoch 3/20
31250/31250 [==============================] - 1294s 41ms/step - loss:
4.1903
Epoch 4/20
31250/31250 [==============================] - 1306s 42ms/step - loss:
4.1875
Epoch 5/20
31250/31250 [==============================] - 1335s 43ms/step - loss:
4.1849
Epoch 6/20
31250/31250 [==============================] - 1345s 43ms/step - loss:
4.1831
Epoch 7/20
31250/31250 [==============================] - 1374s 44ms/step - loss:
4.1821
Epoch 8/20
31250/31250 [==============================] - 1365s 44ms/step - loss:
4.1818
Epoch 9/20
31250/31250 [==============================] - 1374s 44ms/step - loss:
4.1815
Epoch 10/20
31250/31250 [==============================] - 1360s 44ms/step - loss:
4.1814
Epoch 11/20
31250/31250 [==============================] - 1370s 44ms/step - loss:
4.1820
Epoch 12/20
31250/31250 [==============================] - 1386s 44ms/step - loss:
4.1816
Epoch 13/20
31250/31250 [==============================] - 1393s 45ms/step - loss:
4.1807
Epoch 14/20
31250/31250 [==============================] - 1468s 47ms/step - loss:
4.1799
Epoch 15/20
31250/31250 [==============================] - 1447s 46ms/step - loss:
4.1792
Epoch 16/20
31250/31250 [==============================] - 1406s 45ms/step - loss:
4.1793
Epoch 17/20
31250/31250 [==============================] - 1334s 43ms/step - loss:
```

```
4.1786
Epoch 18/20
31250/31250 [==============================] - 1157s 37ms/step - loss:
4.1773
Epoch 19/20
31250/31250 [==============================] - 1035s 33ms/step - loss:
4.1762
Epoch 20/20
31250/31250 [==============================] - 1039s 33ms/step - loss:
4.1754

<keras.callbacks.History at 0x1b614b425f0>

model_lstm = Sequential([
    Embedding(input_dim=vocab_size, output_dim=embedding_dim,
input_length=maxlen),
    LSTM(32, return_sequences=True),
    Dense(vocab_size, activation='softmax')
])

model_lstm.fit(data_pad, summary_pad, epochs=20, batch_size=32)

model_gru = Sequential([
    Embedding(input_dim=vocab_size, output_dim=embedding_dim,
input_length=maxlen),
    GRU(32, return_sequences=True),
    Dense(vocab_size, activation='softmax')
])

model_gru.fit(data_pad, summary_pad, epochs=20, batch_size=32a)

predicted_rnn = model_rnn.predict(data_pad)
predicted_gru = model_gru.predict(data_pad)
predicted_lstm = model_lstm.predict(data_pad)

decoded_rnn = []
decoded_gru = []
decoded_lstm = []
for i in range(len(predicted_rnn)):
    decoded_rnn.append(' '.join([key for key, value in
word_index.items() if np.argmax(predicted_rnn[i]) == value]))
    decoded_gru.append(' '.join([key for key, value in
word_index.items() if np.argmax(predicted_gru[i]) == value]))
    decoded_lstm.append(' '.join([key for key, value in
word_index.items() if np.argmax(predicted_lstm[i]) == value]))

print("RNN Summary:", decoded_rnn)
print("GRU Summary:", decoded_gru)
print("LSTM Summary:", decoded_lstm)
```

```python
references = [[text.split()] for text in data['summary']]
hypotheses_rnn = [text.split() for text in decoded_rnn]
hypotheses_gru = [text.split() for text in decoded_gru]
hypotheses_lstm = [text.split() for text in decoded_lstm]

bleu_rnn = corpus_bleu(references, hypotheses_rnn)
```