nlp-assignment-3

September 25, 2023

[1]:

```
import tensorflow as tf
from tensorflow.keras.layers import Embedding, LSTM, Dense, RepeatVector,␣
 ↪TimeDistributed, Input
from tensorflow.keras.models import Model
from tensorflow.keras.losses import sparse_categorical_crossentropy
import pandas as pd
import re
import string
from string import digits
import numpy as np
```

```
/opt/conda/lib/python3.10/site-packages/scipy/__init__.py:146: UserWarning: A
NumPy version >=1.16.5 and <1.23.0 is required for this version of SciPy
(detected version 1.23.5
  warnings.warn(f"A NumPy version >={np_minversion} and <{np_maxversion}"
```

[2]:

```
df = pd.read_csv("/kaggle/input/english-to-hindi/Hindi_English_Truncated_Corpus.
 ↪csv")
df['source'].value_counts()
```

[2]:

```
source
tides        50000
ted          39881
indic2012    37726
Name: count, dtype: int64
```

[3]:

```
df = df[(df.english_sentence.apply(lambda x: len(str(x)) <= 30)) & (df.
 ↪hindi_sentence.apply(lambda x: len(str(x)) <= 30))]
```

[4]:

```
#Lower each sentence
df['english_sentence'] = df['english_sentence'].apply(lambda x: str(x).lower())
df['hindi_sentence'] = df['hindi_sentence'].apply(lambda x: x.lower())
```

[5]:

```
df['english_sentence'] = df['english_sentence'].apply(lambda x: re.sub("'", '',␣
 ↪x))
df['hindi_sentence'] = df['hindi_sentence'].apply(lambda x: re.sub("'", '', x))
```

[6]:

```
to_exclude = set(string.punctuation)
print("punctuations to exclude:: ", to_exclude)
#Remove Special Characters
df['english_sentence'] = df['english_sentence'].apply(lambda x:''.join(ch for␣
↪ch in x if ch not in to_exclude))

df['hindi_sentence'] = df['hindi_sentence'].apply(lambda x:''.join(ch for ch in␣
↪x if ch not in to_exclude))
```

```
punctuations to exclude::  {')', ']', '&', '_', '>', '[', '?', '}', '/', '!',
'.', '~', "'", '+', '$', '-', '#', '|', '<', '"', ',', '*', '^', '\\', '%', '`',
'=', '{', '(', ':', ';', '@'}
```

[7]:

```
df.head()
```

[7]:

```
          source              english_sentence                  hindi_sentence
11     indic2012      category religious text
23           ted           this changed slowly
26           ted          were being produced
33     indic2012                         maine
35           ted  can you imagine saying that
```

[8]:

```
from string import digits

remove_digits = str.maketrans('', '', digits)
#Remove Digits from the sentences
df['english_sentence'] = df['english_sentence'].apply(lambda x: x.
↪translate(remove_digits))
df['hindi_sentence'] = df['hindi_sentence'].apply(lambda x: x.
↪translate(remove_digits))

df['hindi_sentence'] = df['hindi_sentence'].apply(lambda x: re.sub("[      ]",␣
↪"", x))

#Remove extra spaces
df['english_sentence'] = df['english_sentence'].apply(lambda x: x.strip())
df['hindi_sentence'] = df['hindi_sentence'].apply(lambda x: x.strip())
df['english_sentence'] = df['english_sentence'].apply(lambda x: re.sub(' +', "␣
↪", x))
df['hindi_sentence'] = df['hindi_sentence'].apply(lambda x: re.sub(' +', " ",␣
↪x))
```

[9]:

```
input_text = []
target_text= []
input_characters = set()
target_characters = set()
```

```
for eng, hin in df[['english_sentence', 'hindi_sentence']].
 ↪itertuples(index=False):
  target = 'START_ ' + hin + ' _END'
  input_text.append(eng)
  target_text.append(target)

  for eng_char in eng.split():
    if eng_char not in input_characters:
      input_characters.add(eng_char)

  for hin_char in hin.split():
    if hin_char not in target_characters:
      target_characters.add(hin_char)
```

[10]:

```
print(len(input_text))
print(len(target_text))
print(len(input_characters))
print(len(target_characters))
```

```
18416
18416
9232
8665
```

[11]:

```
from sre_constants import MAX_UNTIL
input_char = sorted(list(input_characters))
target_char = sorted(list(target_characters))

num_encoder_tokens = len(input_characters)
num_decoder_tokens = len(target_characters)

max_encoder_seq_length = max([len(txt) for txt in input_text])
max_decoder_seq_length = max([len(txt) for txt in target_text])
```

[12]:

```
print('Number of samples:', len(input_text))
print('Number of Unique input tokens: ', num_encoder_tokens)
print('Number of Unique output tokens: ', num_decoder_tokens)
print('Max sequence length for inputs: ', max_encoder_seq_length)
print('Max sequence length for outputs: ', max_decoder_seq_length)
```

```
Number of samples: 18416
Number of Unique input tokens:  9232
Number of Unique output tokens:  8665
Max sequence length for inputs:  30
Max sequence length for outputs:  42
```

[13]:

```
input_token_index = dict([(word, i+1) for i, word in enumerate(input_char)])
target_token_index = dict([(word, i+1) for i, word in enumerate(target_char)])
```

[14]:

```
reverse_input_char_index = dict([(i, word) for word, i in input_token_index.
 ↪items()])
reverse_target_char_index = dict([(i, word) for word, i in target_token_index.
 ↪items()])
```

[15]:

```
import pickle
```

[16]:

```
from sklearn.model_selection import train_test_split
X, y = df.english_sentence, df.hindi_sentence
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.1,␣
 ↪random_state = 2)
X_train.shape, X_test.shape
```

[16]:

```
((16574,), (1842,))
```

[17]:

```
def generate_batch(X, y, batch_size):
  while True:
    for j in range(0, len(X), batch_size):
      encoder_input_data = np.zeros((batch_size, max_encoder_seq_length), dtype␣
 ↪= 'float32')
      decoder_input_data = np.zeros((batch_size, max_decoder_seq_length), dtype␣
 ↪= 'float32')
      decoder_target_data = np.zeros((batch_size, max_decoder_seq_length,␣
 ↪num_decoder_tokens), dtype = "float32")

      for i, (input_text, target_text) in enumerate(zip(X[j: j+batch_size],y[j:
 ↪j+batch_size])):
        for t, word in enumerate(input_text.split()):
          encoder_input_data[i, t] = input_token_index[word]

        for t, word in enumerate(target_text.split()):
          if t<len(target_text.split())-1:
            decoder_input_data[i, t] = target_token_index[word]
          if t>0:
            decoder_target_data[i, t-1, target_token_index[word]] = 1.

      yield([encoder_input_data, decoder_input_data], decoder_target_data)
```

[18]:

```
latent_dim = 50
```

[19]:

```
encoder_inputs = Input(shape = (None, ))
enc_emb = Embedding(num_encoder_tokens, latent_dim,␣
 ↪mask_zero=True)(encoder_inputs)
```

```
encoder_lstm = LSTM(latent_dim, return_state=True)
encoder_outputs, state_h, state_c = encoder_lstm(enc_emb)

encoder_states = [state_h, state_c]
```

[20]:
```
decoder_inputs = Input(shape =(None,))
dec_emb_layer = Embedding(num_decoder_tokens, latent_dim, mask_zero = True)
dec_emb = dec_emb_layer(decoder_inputs)

decoder_lstm = LSTM(latent_dim, return_state = True)
decoder_outputs, _, _ = decoder_lstm(dec_emb, initial_state=encoder_states)
decoder_dense = Dense(num_decoder_tokens, activation='softmax')
decoder_outputs = decoder_dense(decoder_outputs)
```

[21]:
```
model= Model([encoder_inputs, decoder_inputs], decoder_outputs)
model.compile(optimizer = 'adam', loss = 'categorical_crossentropy', metrics =␣
 ↪['acc'])
```

[22]:
```
model.summary()
```

```
Model: "model"
________________________________________________________________________________
__________________
 Layer (type)                   Output Shape         Param #     Connected to
================================================================================
==================
 input_1 (InputLayer)           [(None, None)]       0           []

 input_2 (InputLayer)           [(None, None)]       0           []

 embedding (Embedding)          (None, None, 50)     461600
['input_1[0][0]']

 embedding_1 (Embedding)        (None, None, 50)     433250
['input_2[0][0]']

 lstm (LSTM)                    [(None, 50),         20200
['embedding[0][0]']

                                 (None, 50),
                                 (None, 50)]

 lstm_1 (LSTM)                  [(None, 50),         20200
['embedding_1[0][0]',

                                 (None, 50),                      'lstm[0][1]',
                                 (None, 50)]                      'lstm[0][2]']

 dense (Dense)                  (None, 8665)         441915
```

```
['lstm_1[0][0]']

================================================================================
==================
Total params: 1,377,165
Trainable params: 1,377,165
Non-trainable params: 0
--------------------------------------------------------------------------------
------------------
```

[23]:
```
train_samples= len(X_train)
val_samples = len(X_test)
batch_size = 1
epochs = 2
```

[24]:
```
model.fit_generator(generator = generate_batch(X_train, y_train, batch_size =␣
 ↪batch_size),
                    steps_per_epoch = train_samples//batch_size, epochs =␣
 ↪epochs,
                    validation_data = generate_batch(X_test, y_test, batch_size␣
 ↪= batch_size),
                    validation_steps = val_samples//batch_size)
```

```
/tmp/ipykernel_28/3766622033.py:1: UserWarning: `Model.fit_generator` is
deprecated and will be removed in a future version. Please use `Model.fit`,
which supports generators.
  model.fit_generator(generator = generate_batch(X_train, y_train, batch_size =
batch_size),

Epoch 1/2
```

```
---------------------------------------------------------------------------
InvalidArgumentError                      Traceback (most recent call last)
Cell In[24], line 1
----> 1␣
 ↪model.fit_generator(generator = generate_batch(X_train, y_train, batch_size = batch_size),
      2␣
 ↪                    steps_per_epoch = train_samples//batch_size, epochs = epochs,
      3␣
 ↪                    validation_data = generate_batch(X_test, y_test, batch_size = batch_siz
      4                     validation_steps = val_samples//batch_size)

File /opt/conda/lib/python3.10/site-packages/keras/engine/training.py:2636, in␣
 ↪Model.fit_generator(self, generator, steps_per_epoch, epochs, verbose,␣
 ↪callbacks, validation_data, validation_steps, validation_freq, class_weight,␣
 ↪max_queue_size, workers, use_multiprocessing, shuffle, initial_epoch)
   2624 """Fits the model on data yielded batch-by-batch by a Python generator.
   2625
   2626 DEPRECATED:
   2627   `Model.fit` now supports generators, so there is no longer any need t
   2628   use this endpoint.
   2629 """
   2630 warnings.warn(
   2631     "`Model.fit_generator` is deprecated and "
   2632     "will be removed in a future version. "
   2633     "Please use `Model.fit`, which supports generators.",
   2634     stacklevel=2,
   2635 )
-> 2636 return self.fit(
   2637     generator,
   2638     steps_per_epoch=steps_per_epoch,
   2639     epochs=epochs,
   2640     verbose=verbose,
   2641     callbacks=callbacks,
   2642     validation_data=validation_data,
   2643     validation_steps=validation_steps,
   2644     validation_freq=validation_freq,
   2645     class_weight=class_weight,
   2646     max_queue_size=max_queue_size,
   2647     workers=workers,
   2648     use_multiprocessing=use_multiprocessing,
   2649     shuffle=shuffle,
   2650     initial_epoch=initial_epoch,
   2651 )

File /opt/conda/lib/python3.10/site-packages/keras/utils/traceback_utils.py:70,
 ↪in filter_traceback.<locals>.error_handler(*args, **kwargs)
     67     filtered_tb = _process_traceback_frames(e.__traceback__)
     68     # To get the full stack trace, call:
     69     # `tf.debugging.disable_traceback_filtering()`
---> 70     raise e.with_traceback(filtered_tb) from None
     71 finally:
     72     del filtered_tb

File /opt/conda/lib/python3.10/site-packages/tensorflow/python/eager/execute.py
 ↪52, in quick_execute(op_name, num_outputs, inputs, attrs, ctx, name)
     50 try:
     51   ctx.ensure_initialized()
---> 52   tensors = pywrap_tfe.TFE_Py_Execute(ctx._handle, device_name, op_name
     53                                       inputs, attrs, num_outputs)
     54 except core._NotOkStatusException as e:
     55   if name is not None:

InvalidArgumentError: Graph execution error:
```

```
Detected at node 'categorical_crossentropy/remove_squeezable_dimensions/Squeeze ␣
 ↪defined at (most recent call last):
    File "/opt/conda/lib/python3.10/runpy.py", line 196, in _run_module_as_main
      return _run_code(code, main_globals, None,
    File "/opt/conda/lib/python3.10/runpy.py", line 86, in _run_code
      exec(code, run_globals)
    File "/opt/conda/lib/python3.10/site-packages/ipykernel_launcher.py", line␣
 ↪17, in <module>
      app.launch_new_instance()
    File "/opt/conda/lib/python3.10/site-packages/traitlets/config/application.
 ↪py", line 1043, in launch_instance
      app.start()
    File "/opt/conda/lib/python3.10/site-packages/ipykernel/kernelapp.py", line
 ↪728, in start
      self.io_loop.start()
    File "/opt/conda/lib/python3.10/site-packages/tornado/platform/asyncio.py",
 ↪line 195, in start
      self.asyncio_loop.run_forever()
    File "/opt/conda/lib/python3.10/asyncio/base_events.py", line 603, in␣
 ↪run_forever
      self._run_once()
    File "/opt/conda/lib/python3.10/asyncio/base_events.py", line 1909, in␣
 ↪_run_once
      handle._run()
    File "/opt/conda/lib/python3.10/asyncio/events.py", line 80, in _run
      self._context.run(self._callback, *self._args)
    File "/opt/conda/lib/python3.10/site-packages/ipykernel/kernelbase.py", lin ␣
 ↪513, in dispatch_queue
      await self.process_one()
    File "/opt/conda/lib/python3.10/site-packages/ipykernel/kernelbase.py", lin ␣
 ↪502, in process_one
      await dispatch(*args)
    File "/opt/conda/lib/python3.10/site-packages/ipykernel/kernelbase.py", lin ␣
 ↪409, in dispatch_shell
      await result
    File "/opt/conda/lib/python3.10/site-packages/ipykernel/kernelbase.py", lin ␣
 ↪729, in execute_request
      reply_content = await reply_content
    File "/opt/conda/lib/python3.10/site-packages/ipykernel/ipkernel.py", line␣
 ↪422, in do_execute
      res = shell.run_cell(
    File "/opt/conda/lib/python3.10/site-packages/ipykernel/zmqshell.py", line␣
 ↪540, in run_cell
      return super().run_cell(*args, **kwargs)
    File "/opt/conda/lib/python3.10/site-packages/IPython/core/interactiveshell
 ↪py", line 3009, in run_cell
      result = self._run_cell(
  File "/opt/conda/lib/python3.10/site-packages/IPython/core/interactiveshell
↪py", line 3064, in _run_cell
    result = runner(coro)
  File "/opt/conda/lib/python3.10/site-packages/IPython/core/async_helpers.
↪py", line 129, in _pseudo_sync_runner
    coro.send(None)
  File "/opt/conda/lib/python3.10/site-packages/IPython/core/interactiveshell
↪py", line 3269, in run_cell_async
    has_raised = await self.run_ast_nodes(code_ast.body, cell_name,
  File "/opt/conda/lib/python3.10/site-packages/IPython/core/interactiveshell
↪py", line 3448, in run_ast_nodes
    if await self.run_code(code, result, async_=asy):
  File "/opt/conda/lib/python3.10/site-packages/IPython/core/interactiveshell
↪py", line 3508, in run_code
    exec(code_obj, self.user_global_ns, self.user_ns)
  File "/tmp/ipykernel_28/3766622033.py", line 1, in <module>
    model.fit_generator(generator = generate_batch(X_train, y_train,␣
↪batch_size = batch_size),
  File "/opt/conda/lib/python3.10/site-packages/keras/engine/training.py",␣
↪line 2636, in fit_generator
    return self.fit(
  File "/opt/conda/lib/python3.10/site-packages/keras/utils/traceback_utils.
↪py", line 65, in error_handler
    return fn(*args, **kwargs)
  File "/opt/conda/lib/python3.10/site-packages/keras/engine/training.py",␣
↪line 1685, in fit
    tmp_logs = self.train_function(iterator)
  File "/opt/conda/lib/python3.10/site-packages/keras/engine/training.py",␣
↪line 1284, in train_function
    return step_function(self, iterator)
  File "/opt/conda/lib/python3.10/site-packages/keras/engine/training.py",␣
↪line 1268, in step_function
    outputs = model.distribute_strategy.run(run_step, args=(data,))
  File "/opt/conda/lib/python3.10/site-packages/keras/engine/training.py",␣
↪line 1249, in run_step
    outputs = model.train_step(data)
  File "/opt/conda/lib/python3.10/site-packages/keras/engine/training.py",␣
↪line 1051, in train_step
    loss = self.compute_loss(x, y, y_pred, sample_weight)
  File "/opt/conda/lib/python3.10/site-packages/keras/engine/training.py",␣
↪line 1109, in compute_loss
    return self.compiled_loss(
  File "/opt/conda/lib/python3.10/site-packages/keras/engine/compile_utils.
↪py", line 265, in __call__
    loss_value = loss_obj(y_t, y_p, sample_weight=sw)
  File "/opt/conda/lib/python3.10/site-packages/keras/losses.py", line 142, in␣
↪__call__
    losses = call_fn(y_true, y_pred)
    File "/opt/conda/lib/python3.10/site-packages/keras/losses.py", line 261, i␣
  ↪call
      y_pred, y_true = losses_utils.squeeze_or_expand_dimensions(
    File "/opt/conda/lib/python3.10/site-packages/keras/utils/losses_utils.py",␣
  ↪line 200, in squeeze_or_expand_dimensions
      y_true, y_pred = remove_squeezable_dimensions(y_true, y_pred)
    File "/opt/conda/lib/python3.10/site-packages/keras/utils/losses_utils.py",␣
  ↪line 139, in remove_squeezable_dimensions
      labels = tf.squeeze(labels, [-1])
Node: 'categorical_crossentropy/remove_squeezable_dimensions/Squeeze'
Can not squeeze dim[2], expected a dimension of 1, got 8665
         [[{{node categorical_crossentropy/remove_squeezable_dimensions/
  ↪Squeeze}}]] [Op:__inference_train_function_14275]
```

INFERENCE

[ ]:
```
model.save_weights('nmt_eng_hin_translation.h5')
```

[ ]:
```
encoder_model = Model(encoder_inputs, encoder_states)
```

[ ]:
```
decoder_state_input_h = Input(shape=(latent_dim, ))
decoder_state_input_c = Input(shape=(latent_dim, ))
decoder_state_inputs = [decoder_state_input_h, decoder_state_input_c]
```

[ ]:
```
dec_emb2 = dec_emb_layer(decoder_inputs)
```

[ ]:
```
decoder_outputs2, state_h2, state_c2 = decoder_lstm(dec_emb2,␣
 ↪initial_state=decoder_state_inputs)
decoder_states2 = [state_h2, state_c2]
decoder_outputs2 = decoder_dense(decoder_outputs2)
```

[ ]:
```
decoder_model = Model([decoder_inputs] + decoder_state_inputs,␣
 ↪[decoder_outputs2] + decoder_states2)
```

[ ]:
```
def decoder_sequence(input_seq):
  states_value = encoder_model.predict(input_seq)
  target_seq = np.zeros((1,1))
  #Populate the first character of the target sequence with the start character.
  #target_seq[0,0] = target_token_index['START_ ']

  stop_condition = False
  decoded_sequence = ''

  while not stop_condition:
    output_tokens, h, c = decoder_model.predict([target_seq] + states_value)

    sampled_token_index = np.argmax(output_tokens[0, -1, : ])
    sampled_char = reverse_target_char_index[sampled_token_index]
    decoded_sentence += ' ' + sampled_char

    if(sampled_char == ' _END' or len(decoded_sentence) > 25):
      stop_condition = True

    target_seq = np.zeros((1,1))
    target_seq[0,0] = sampled_token-index

    states_value = [h,c]

  return decoded_sequence
```

[ ]:

```
val_gen = generate_batch(X_test, y_test, batch_size = 1)
k=-1
```

[ ]:

```
k+=2
(input_seq, actual_output), _ = next(val_gen)
decoded_sentence = decode_sequence(input_seq)
print('Input English sentence:', X_test[k:k+1].values[0])
print('Actual Hindi Translation:', y_test[k:k+1].values[0])
print('Predicted Hindi Translation:', decoded_sentence[:])
```

[1]:

```
pip install nbconvert
```

```
Requirement already satisfied: nbconvert in
/Users/harshgandhi/opt/anaconda3/lib/python3.9/site-packages (6.4.4)
Requirement already satisfied: defusedxml in
/Users/harshgandhi/opt/anaconda3/lib/python3.9/site-packages (from nbconvert)
(0.7.1)
Requirement already satisfied: pandocfilters>=1.4.1 in
/Users/harshgandhi/opt/anaconda3/lib/python3.9/site-packages (from nbconvert)
(1.5.0)
Requirement already satisfied: beautifulsoup4 in
/Users/harshgandhi/opt/anaconda3/lib/python3.9/site-packages (from nbconvert)
(4.11.1)
Requirement already satisfied: testpath in
/Users/harshgandhi/opt/anaconda3/lib/python3.9/site-packages (from nbconvert)
(0.6.0)
Requirement already satisfied: mistune<2,>=0.8.1 in
/Users/harshgandhi/opt/anaconda3/lib/python3.9/site-packages (from nbconvert)
(0.8.4)
Requirement already satisfied: jupyterlab-pygments in
/Users/harshgandhi/opt/anaconda3/lib/python3.9/site-packages (from nbconvert)
(0.1.2)
Requirement already satisfied: jupyter-core in
/Users/harshgandhi/opt/anaconda3/lib/python3.9/site-packages (from nbconvert)
```

```
(4.11.1)
Requirement already satisfied: nbformat>=4.4 in
/Users/harshgandhi/opt/anaconda3/lib/python3.9/site-packages (from nbconvert)
(5.5.0)
Requirement already satisfied: entrypoints>=0.2.2 in
/Users/harshgandhi/opt/anaconda3/lib/python3.9/site-packages (from nbconvert)
(0.4)
Requirement already satisfied: bleach in
/Users/harshgandhi/opt/anaconda3/lib/python3.9/site-packages (from nbconvert)
(4.1.0)
Requirement already satisfied: nbclient<0.6.0,>=0.5.0 in
/Users/harshgandhi/opt/anaconda3/lib/python3.9/site-packages (from nbconvert)
(0.5.13)
Requirement already satisfied: traitlets>=5.0 in
/Users/harshgandhi/opt/anaconda3/lib/python3.9/site-packages (from nbconvert)
(5.1.1)
Requirement already satisfied: pygments>=2.4.1 in
/Users/harshgandhi/opt/anaconda3/lib/python3.9/site-packages (from nbconvert)
(2.11.2)
Requirement already satisfied: jinja2>=2.4 in
/Users/harshgandhi/opt/anaconda3/lib/python3.9/site-packages (from nbconvert)
(2.11.3)
Requirement already satisfied: MarkupSafe>=0.23 in
/Users/harshgandhi/opt/anaconda3/lib/python3.9/site-packages (from
jinja2>=2.4->nbconvert) (2.0.1)
Requirement already satisfied: jupyter-client>=6.1.5 in
/Users/harshgandhi/opt/anaconda3/lib/python3.9/site-packages (from
nbclient<0.6.0,>=0.5.0->nbconvert) (7.3.4)
Requirement already satisfied: nest-asyncio in
/Users/harshgandhi/opt/anaconda3/lib/python3.9/site-packages (from
nbclient<0.6.0,>=0.5.0->nbconvert) (1.5.5)
Requirement already satisfied: jsonschema>=2.6 in
/Users/harshgandhi/opt/anaconda3/lib/python3.9/site-packages (from
nbformat>=4.4->nbconvert) (4.16.0)
Requirement already satisfied: fastjsonschema in
/Users/harshgandhi/opt/anaconda3/lib/python3.9/site-packages (from
nbformat>=4.4->nbconvert) (2.16.2)
Requirement already satisfied: soupsieve>1.2 in
/Users/harshgandhi/opt/anaconda3/lib/python3.9/site-packages (from
beautifulsoup4->nbconvert) (2.3.1)
Requirement already satisfied: webencodings in
/Users/harshgandhi/opt/anaconda3/lib/python3.9/site-packages (from
bleach->nbconvert) (0.5.1)
Requirement already satisfied: six>=1.9.0 in
/Users/harshgandhi/opt/anaconda3/lib/python3.9/site-packages (from
bleach->nbconvert) (1.16.0)
Requirement already satisfied: packaging in
/Users/harshgandhi/opt/anaconda3/lib/python3.9/site-packages (from
bleach->nbconvert) (21.3)
Requirement already satisfied: attrs>=17.4.0 in
/Users/harshgandhi/opt/anaconda3/lib/python3.9/site-packages (from
jsonschema>=2.6->nbformat>=4.4->nbconvert) (21.4.0)
Requirement already satisfied: pyrsistent!=0.17.0,!=0.17.1,!=0.17.2,>=0.14.0 in
/Users/harshgandhi/opt/anaconda3/lib/python3.9/site-packages (from
jsonschema>=2.6->nbformat>=4.4->nbconvert) (0.18.0)
Requirement already satisfied: python-dateutil>=2.8.2 in
/Users/harshgandhi/opt/anaconda3/lib/python3.9/site-packages (from jupyter-
client>=6.1.5->nbclient<0.6.0,>=0.5.0->nbconvert) (2.8.2)
Requirement already satisfied: tornado>=6.0 in
/Users/harshgandhi/opt/anaconda3/lib/python3.9/site-packages (from jupyter-
client>=6.1.5->nbclient<0.6.0,>=0.5.0->nbconvert) (6.1)
Requirement already satisfied: pyzmq>=23.0 in
/Users/harshgandhi/opt/anaconda3/lib/python3.9/site-packages (from jupyter-
client>=6.1.5->nbclient<0.6.0,>=0.5.0->nbconvert) (23.2.0)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in
/Users/harshgandhi/opt/anaconda3/lib/python3.9/site-packages (from
packaging->bleach->nbconvert) (3.0.9)
Note: you may need to restart the kernel to use updated packages.
```

[3]:

```
!ipython nbconvert harshgandhi-nlpassignment-machinetranslation.ipynb␣
 ↪--to=latex --post=PDF
```

```
zsh:1: command not found: ipython
```

[ ]:

```
```